

Implementació d'un sistema de gestió i alarma en una xarxa comunitària sense fils

Autor: Sergi Pica Stauss

3/7/2018

Director: Llorenç Cerdà-Alabern – UPC AC

Grau en Enginyeria Informàtica

Tecnologies de la Informació

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Barcelona(UPC) – Barcelona Tech



Resum

Avui dia els dispositius WiFi tenen un preu molt baix, cosa que ha fet que nombroses comunitats sense fils s'hagin desplegat. Un exemple d'aquestes és la xarxa de Guifi·Sants, que és una xarxa mesh sense fils desplegada al barri de Sants, a Barcelona.

Aquesta xarxa permet a que els seus nodes que en formen part es comuniquin entre si i aconseguixin connectar-se a Internet.

Al dependre de altres usuaris (nodes) per a que la xarxa funcioni correctament , és important saber l'estat de tots els nodes en tot moment. Així doncs, en aquest projecte es desenvoluparà un sistema de gestió i alarma en cas de fallada dels nodes en aquesta xarxa, és a dir, es controlarà la disponibilitat de cadascun d'aquests nodes de forma periòdica i en cas de detectar algun problema, es generarà una notificació automàtica per a informar d'aquest estat.

Glossari

Xarxa mesh: és una xarxa on tots els nodes tenen diverses connexions entre si, de manera que la fallada d'un, no ha de repercutir necessàriament en el funcionament de la xarxa

Node: són cadascun dels usuaris que tenen un dispositiu (antena) connectat a la xarxa.

Guifi·net: guifi-net és un projecte tecnològic, social i econòmic impulsat des de la ciutadania que té per objectiu la creació d'una xarxa de telecomunicacions oberta, lliure i neutral basada en un model de comuns.

Guifi·Sants: és una comunitat de guifi-net situada al barri de Sants .

Arquitectura QMPSU: és la arquitectura que s'utilitza a la xarxa de Guifi·Sants.

Abstract

Nowadays Wi-Fi devices have a low cost, and thus, lots of wireless communities have been deployed. A good example of these is Guifi-Sants, a wireless mesh network deployed in the quarter of Sants, Barcelona.

This network allows its nodes to communicate and get an Internet connection.

As you depend on the other users (nodes) for the network to work properly, it's important to know the state of the nodes at all time. So, in this project, we will develop a management and alarm system in case of a failure of the network's nodes. So, it must control the availability of every node periodically and, if an error is detected, the system will generate an automatic notification to inform of the state.

Glossary

Mesh network: *a network where all the nodes have several connections between them, so that the failure of one node doesn't necessarily mean that the network stops working.*

Node: *every one of the user's devices (antennas) connected to the network.*

Guifi-net: *guifi-net is a bottom-up, citizenship-driven technological, social and economic project with the objective of creating a free, open and neutral telecommunications network based on a commons model.*

Guifi-Sants: *it's a guifi-net community situated in the Sants quarter in Barcelona*

QMPSU Architecture: *the architecture used in Guifi-Sants' network.*

Resumen

Hoy en día los dispositivos Wifi tienen un precio muy bajo, lo cual ha hecho que muchas comunidades inalámbricas se hayan desplegado. Un ejemplo de ellas es la red de Guifi·Sants, que es una red mesh inalámbrica que se sitúa en el barrio de Sants, en Barcelona.

Esta red permite a los nodos que la forman que se comuniquen entre sí y lleguen a tener conexión a Internet.

Al depender de otros usuarios (nodos) para que la red funcione correctamente, es necesario saber el estado de los nodos en cualquier momento. Así pues, en este proyecto se desarrollará un sistema de gestión y alarma en caso de algún fallo en los nodos de la red, es decir, se controlará la disponibilidad de cada uno de los nodos de forma periódica y en caso de detectar algún problema, se generará una notificación automática para informar del estado.

Glosario

Red mesh: es una red donde todos los nodos tienen distintas conexiones entre sí, de forma que un fallo en uno, no tiene que repercutir necesariamente en el funcionamiento de la red.

Nodo: cada uno de los usuarios que tienen un dispositivo (antena) conectado a la red.

Guifi·net: es un proyecto tecnológico, social y económico impulsado desde la ciudadanía que tiene por objetivo la creación de una red de telecomunicaciones abierta, libre y neutral basada en un modelo de procomún.

Guifi·Sants: es una comunidad de guifi·net situada en el barrio de Sants .

Arquitectura QMPSU: es la arquitectura que se usa en la red de Guifi·Sants.

Índex

1. CONTEXT	1
1.1. INTRODUCCIÓ	1
1.2. ACTORS IMPLICATS	2
1.2.1. <i>Desenvolupador</i>	2
1.2.2. <i>Director</i>	2
1.2.3. <i>Beneficiaris</i>	2
2. ESTAT DE L'ART	3
2.1. XARXES WMN	3
2.1.1. <i>Guifi.net</i>	3
2.1.2. <i>MadMesh</i>	4
2.1.3. <i>Google WiFi</i>	4
2.1.4. <i>Meraki</i>	5
2.2. ARQUITECTURA QMPSU	5
2.3. CONCLUSIONS	5
3. FORMULACIÓ DEL PROBLEMA	6
3.1. OBJECTIUS	7
4. ABAST	8
4.1. META FINAL	8
4.2. POSSIBLES OBSTACLES	9
4.2.1. <i>Errors de programació</i>	9
4.2.2. <i>Manca de coneixements</i>	9
4.2.3. <i>Calendari</i>	9
5. PLANIFICACIÓ TEMPORAL	10
5.1. DESCRIPCIÓ DE LES TASQUES	10
5.1.1. <i>Fita inicial</i>	10
5.1.2. <i>Estudi de les eines</i>	10
5.1.3. <i>Implementació</i>	11
5.1.4. <i>Integració de les aplicacions</i>	11
5.1.5. <i>Documentació</i>	11
5.2. DEPENDÈNCIES	11
5.3. PLA D'ACCIÓ I VALORACIÓ D'ALTERNATIVES	12
5.4. CALENDARI DEL PROJECTE	13
5.4.1. <i>Estimació d'hores per tasca</i>	13
5.4.2. <i>Diagrama de Gantt</i>	14
5.5. IMPREVISTOS	15
6. RECURSOS	16
6.1. RECURSOS PERSONALS	16
6.2. RECURSOS MATERIALS	16
6.2.1. <i>Hardware</i>	16
6.2.2. <i>Software</i>	17
7. GESTIÓ ECONÒMICA DEL PROJECTE	18
7.1. ESTIMACIÓ DELS COSTOS	18

7.1.1.	<i>Costos de recursos humans</i>	18
7.1.2.	<i>Costos de recursos materials</i>	19
7.1.3.	<i>Costos generals</i>	20
7.1.4.	<i>Imprevistos</i>	20
7.1.5.	<i>Contingència</i>	21
7.1.6.	<i>Pressupost final</i>	21
7.2.	CONTROL DE GESTIÓ	21
8.	METODOLOGIA I RIGOR	23
8.1.	FITA INICIAL	23
8.1.1.	<i>Mètode de treball</i>	23
8.1.2.	<i>Eines</i>	23
8.1.3.	<i>Entorn de treball</i>	24
8.1.4.	<i>Mètode de validació</i>	25
9.	DESENVOLUPAMENT I IMPLEMENTACIÓ	26
9.1.	NAGIOS	26
9.1.1.	<i>Ús en el projecte</i>	27
9.1.2.	<i>Consideracions tècniques</i>	27
9.1.3.	<i>Instal·lació</i>	32
9.1.4.	<i>Servidor de correu</i>	36
9.2.	IMPLEMENTACIÓ DE INTERFÍCIE WEB	40
9.2.1.	<i>Instal·lació</i>	40
9.2.2.	<i>Web</i>	41
9.2.3.	<i>CGI</i>	42
9.3.	IMPLEMENTACIÓ DE BASE DE DADES	43
9.3.1.	<i>Instal·lació</i>	43
9.3.2.	<i>Configuració</i>	45
9.4.	INTEGRACIÓ DE LES TECNOLOGIES	47
9.4.1.	<i>Connexió web – base de dades</i>	47
9.4.2.	<i>Connexió base de dades – Nagios</i>	49
10.	SOSTENIBILITAT I COMPROMÍS SOCIAL	53
10.1.	AUTOAVALUACIÓ	53
10.2.	DIMENSIÓ AMBIENTAL	53
10.3.	DIMENSIÓ ECONÒMICA	54
10.4.	DIMENSIÓ SOCIAL	54
11.	CONCLUSIONS	56
	REFERÈNCIES	57
	ANNEXOS	58
	ANNEX A JAVASCRIPT PER OBTENIR DADES DEL JSON	58
	ANNEX B CODI DEL CGI	59
	ANNEX C SCRIPT PER A CREAR LES COMANDES DE NAGIOS	64
	ANNEX D SCRIPT PER CREAR ELS SERVEIS DE NAGIOS	66

Índex de figures

Figura 1 Esquema d'una xarxa Mesh	2
Figura 2 Model de Guifi.net.....	4
Figura 3 Captura de la pàgina de monitorització de xarxes Mesh qMP.....	7
Figura 4 Exemple de la interfície gràfica de Nagios.....	8
Figura 5 Diagrama de Gantt.....	14
Figura 6 Graf representatiu de la metodologia cascada	23
Figura 7 Interfície web de Nagios	28
Figura 8 Vista dels serveis de Nagios en la interfície web	28
Figura 9 Sortida de terminal de la configuració	33
Figura 10 Sortida de terminal al realitzar la compilació de Nagios	33
Figura 11 Pestanya de serveis de Nagios sense cap servei actiu	35
Figura 12 Pestanya de serveis de Nagios amb serveis d'exemple actius	36
Figura 13 Correu generat per emailcheck	39
Figura 14 Pàgina per defecte de apache	41
Figura 15 Interfície de la base de dades	44
Figura 16 Taula NodeIP plena.....	47
Figura 17 Exemple de sortida al registrar-se	49

Índex de taules

Taula 1 Dependències entre tasques	12
Taula 2 Planificació temporal de les tasques	13
Taula 3 Temps dedicat per tasca real	15
Taula 4 Recursos Hardware	16
Taula 5 Recursos Software	17
Taula 6 Cost de recursos humans.....	18
Taula 7 Cost de recursos humans segons activitats	19
Taula 8 Costos indirectes per recursos materials hardware	20
Taula 9 Costos generals	20
Taula 10 Costos d'imprevistos.....	21
Taula 11 Costos de contingència	21
Taula 12 Pressupost final.....	21

1. Context

1.1. Introducció

El baix cost dels dispositius Wi-Fi ha portat a que sorgeixin diverses xarxes mesh sense fils [1]. Aquest fet ha afavorit que molts aficionats i voluntaris creïn les seves pròpies xarxes comunitàries sense fils, que destaquen pel fet de que la organització i el desplegament d'aquesta xarxa la realitzen els mateixos usuaris, i així, al contrari que en el sistema tradicional de operadores de telecomunicacions, tots els usuaris son els propietaris d'una part de la infraestructura total. Usant un bon sistema d'organització, els usuaris es poden connectar als seus veïns (i veïns de veïns). Aquestes xarxes acostumen a ser lliures i obertes [2] [3].

Un exemple d'aquestes xarxes és el de Guifi.net. El Guifi.net és la xarxa comunitària més gran que existeix actualment i disposa de aproximadament 32.500 nodes actius i es calcula que més de 50.000 persones reben servei a través de la xarxa de comuns [4]. Es va començar a utilitzar per a donar accés a Internet a zones rurals on les operadores tradicionals no tenien interès en fer arribar els seus serveis. Un factor important del seu èxit és que molts cops s'utilitza per organitzar petites comunitats, les quals tenen el seu propi hardware, software i mètodes d'organització. Però tots ells comparteixen la part més important de Guifi.net, la seva pàgina web, que s'utilitza per distribuir les IPs i ajuntar aquestes comunitats amb un sistema comú.

Concretament, aquest treball es centra en una comunitat de Guifi.net, Guifi·Sants [5]. Aquesta comunitat neix al 2009 com iniciativa popular per estendre al barri de Sants la xarxa alternativa, lliure i neutral Guifi·net. Aquesta comunitat s'organitza amb el model de xarxa en malla (mesh)(Figura 1), on cada client és també un node repetidor.

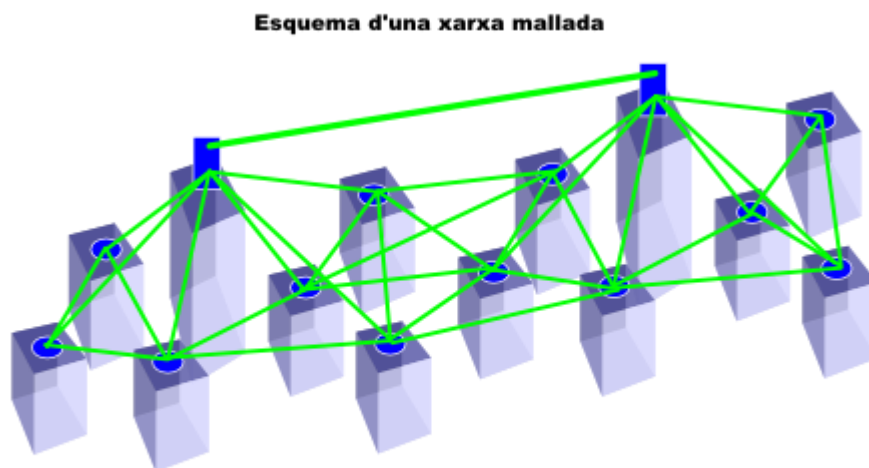


Figura 1 Esquema d'una xarxa Mesh

1.2. Actors implicats

1.2.1. Desenvolupador

És la persona que s'encarrega de documentar, fer recerca i implementar la solució al problema que s'especifica. També és el responsable de la gestió del projecte i de escriure tota la documentació necessària. Aquest actor treballa amb el que s'acorda amb el director i és el responsable de les dades límit.

1.2.2. Director

És l'encarregat de guiar, donar consell i ajudar al desenvolupador. És imprescindible per a detectar errors en el projecte i ajudar a corregir-los el més aviat possible. Per aquest projecte, el director és Llorenç Cerdà-Alabern.

1.2.3. Beneficiaris

Els beneficiaris d'aquest projecte són aquelles persones o entitats que poden utilitzar el resultat del projecte a favor seu. En aquest cas, són tots els usuaris que formen part de Guifi·Sants, ja que podran registrar-se i obtenir els avisos i alertes sobre l'estat de la seva xarxa.

2. Estat de l'art

Actualment, els preus dels dispositius WiFi ha disminuït molt i amb la sortida d'estàndards nous com el 802.11n, se n'ha multiplicat el seu rendiment. Això ha fet que s'hagi fomentat el desplegament de xarxes mesh sense fils, posant en pràctica conceptes que venen de la recerca de xarxes Ad Hoc. Tot i que és un tema relativament nou, els escenaris en que s'utilitzen xarxes mesh sense fils és cada dia més gran. [2] [3]

Dit això, s'ha fet molta recerca en els darrers anys relacionat amb aquestes xarxes, incloent aspectes de disseny (routing, escalabilitat i seguretat), desplegament, mesures i anàlisi, rendiment, ús, etc.

A continuació s'exposen diverses xarxes WMN (Wireless Mesh networks) que s'han desplegat de forma similar a la xarxa sobre la qual es realitza el projecte.

2.1. Xarxes WMN

2.1.1. Guifi.net

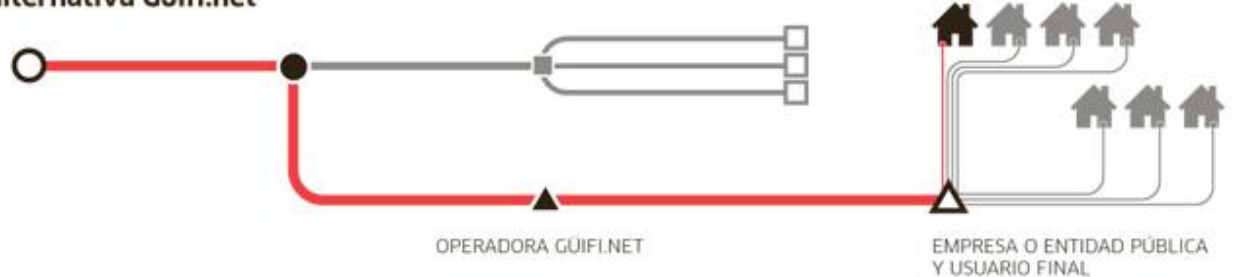
Guifi.net [4] [1] és una xarxa comunitària sense fils que es va originar l'any 2004. Guifi.net és un projecte tecnològic, social i econòmic impulsat des de la ciutadania que té per objectiu la creació d'una xarxa de telecomunicacions oberta, lliure i neutral basada en un model de comuns.

En la Figura 2, podem veure un esquema en el qual hi ha representats els models clàssics de les telefòniques i també es pot veure la alternativa que proporciona Guifi per a proporcionar els serveis als seus usuaris.

El modelo clásico basado en las telefónicas



La alternativa Güifi.net



1

Figura 2 Model de Guifi.net

Guifi.net és una xarxa molt heterogènia, majoritàriament en estructura, ja que molts dels seus links s'estableixen manualment i utilitzen OSPF i BGP com a protocols de Routing. Tot i això, recentment s'han afegit xarxes mesh, com la de Sants, que utilitza la arquitectura qMPSU i és sobre la qual es realitza el treball.

2.1.2. MadMesh

MadMesh és una WMN desplegada de forma planificada per CISCO l'any 2007. S'usa per a proporcionar accés d'Internet a petites empreses i particulars a Madison, Wisconsin. L'estudi que s'hi va realitzar consistia en 8 mesos de recol·lecció de dades usant logs SNMP. Els autors van fer uns amplis anàlisis que cobrien aspectes topològics de robustesa i activitat dels usuaris, entre d'altres.

2.1.3 Google WiFi

Google WiFi [6] és un altre exemple de WMN desplegada de forma planificada que es va realitzar a Mountain View, California a l'any 2008. Aquesta xarxa està basada en la WMN

¹ Figura disponible a: <https://elx.guifi.net/wp-content/uploads/mapa-red-guifi.jpg>

propietària desenvolupada per Tropos (MetroMesh). L'objectiu d'aquesta xarxa és la recerca, i ofereix Internet gratuït als usuaris. El objectiu principal és derivar patrons d'ús de la activitat dels clients.

2.1.4. Meraki

Meraki [7] era una companyia que van crear S. Iswas i J. Bicket basada en part en el projecte MIT Roofnet, i va ser comprada per CISCO al 2012. La xarxa Meraki fa desplegaments comercials de WMN centrats en petites organitzacions com escoles, hotels o hospitals. Sobre aquesta xarxa s'han fet estudis sobre mesures a nivell de link, l'impacte d'estacions ocultes i comparant el SNR contra el bit-rate.

2.2. Arquitectura QMPSU

La arquitectura QMPSU [8] és una arquitectura basada en WMN 802.11an, que es va iniciar el 2011 al barri de Sants de Barcelona. La majoria dels usuaris d'aquesta xarxa la usen com a mètode per a obtenir accés a Internet i depenen de la comunitat. Per tant, és necessària una bona estabilitat i rendiment per a aquesta xarxa. Aquesta arquitectura és la que s'utilitza en la xarxa de Guifi·Sants [5] i és sobre la que es realitzarà el projecte.

2.3. Conclusions

Per concloure, avui dia hi ha molta recerca i moltes implementacions distintes de xarxes mesh sense fils. Tot i que pot ser difícil de caracteritzar unes xarxes tan dinàmiques, els resultats mostren que les xarxes normalment estan prou ben connectades i són adaptables. Amb això es demostra que aquestes xarxes aporten un gran nombre de avantatges respecte a les connexions convencionals.

3. Formulació del problema

Actualment, a la xarxa mesh de Guifi-Sants, hi ha un total de 87 nodes. Aquests nodes no són tots iguals, tenen hardwares diferents i alhora tenen un nombre de connexions diferents cadascun.

Un usuari, per a poder connectar-se a aquesta xarxa comunitària necessita (per a la seva versió més bàsica) una petita antena wifi d'exterior, que es sol col·locar als teulats dels edificis i un cable d'Ethernet que permet alimentar l'antena i transportar les dades a un router domèstic. També es fa ús del Quick Mesh Project (qMP), que és un firmware per a dispositius de xarxa (per exemple antenes) que permet d'una manera senzilla formar part de la Mesh de Sants.

Tenint en compte aquests factors, veiem que la xarxa en qüestió és una xarxa ben heterogènia, i, per tant, poden sorgir certs problemes de connexió.

Una de les formes que hi ha per comprovar l'estat de la xarxa, és a través de una web (realitzada i mantinguda per el professor Llorenç Cerdà-Alabern) que monitoritza i mostra de forma visual dades relacionades amb el tràfic de dades que hi ha en cadascun dels nodes existents connectats a la Mesh, tal i com es pot veure en la Figura 3. En aquesta web es pot veure per cada node les connexions que té amb altres nodes, així com el tipus d'antena, el seu nom, direccions IP i molts altres paràmetres de xarxa que es van actualitzant de forma periòdica cada hora. També s'hi mostra una representació gràfica de tots els links així com grafs amb distintes informacions (distàncies entre nodes, nodes actius, etc).

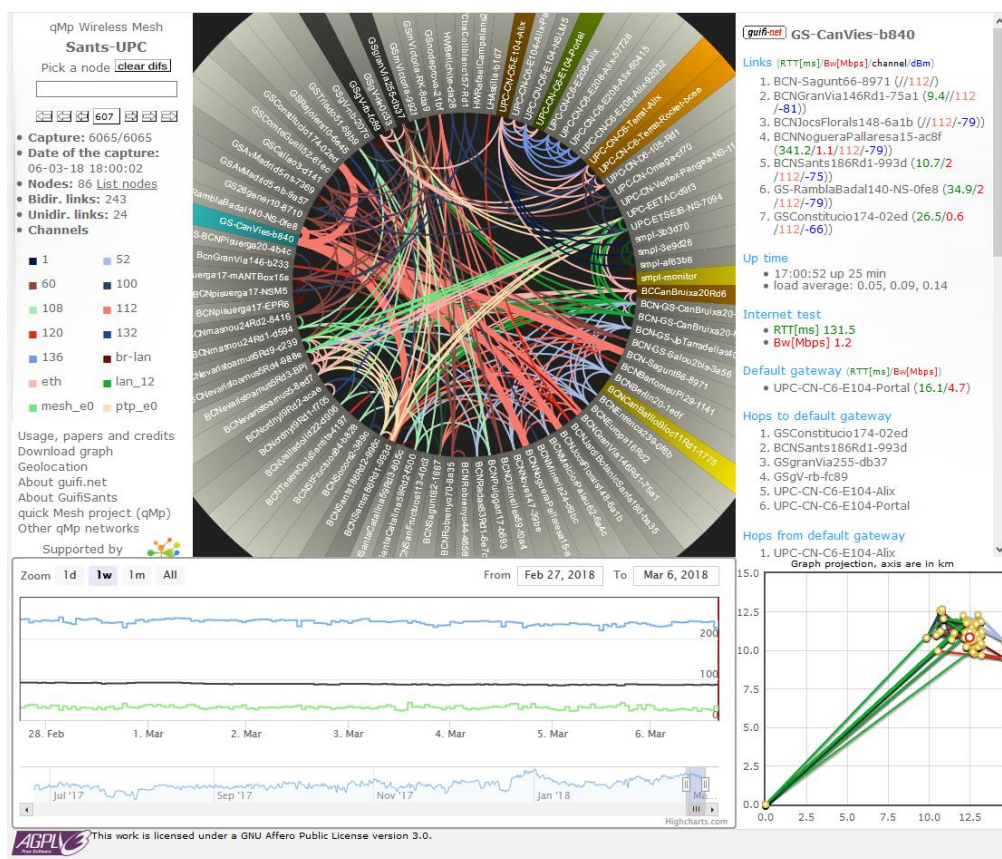


Figura 3 Captura de la pàgina de monitorització de xarxes Mesh qMp²

En el cas de que es produeixi un error en la xarxa, pot ser difícil de detectar i per tant, trobar una solució encertada i ràpida.

3.1. Objectius

Així doncs, partint de les característiques i problemes descrits anteriorment, els objectius que s'intenten implementar amb aquest Treball de Fi de Grau són:

- Analitzar quines de les dades relacionades amb el tràfic de la xarxa són les més rellevants per a detectar si hi ha alguna fallada en el sistema.
- Dissenyar i implementar un sistema de monitorització i alarma en xarxa, que tingui en compte el estat de la xarxa i detecti les fallades que s'hi produeixin.
- Aconseguir que els usuaris interessats s'hi puguin registrar i sol·licitar les alarmes que els siguin més útils. En cas de que es detecti la fallada que interessa a l'usuari, el sistema de monitorització haurà d'enviar un correu electrònic als usuaris que s'hi hagin subscrit.

² Figura corresponent a la web: <http://dsg.ac.upc.edu/qmpsu/>

4. Abast

4.1. Meta final

L'abast d'aquest projecte és realitzar i desenvolupar un sistema d'alarma que s'encarregui d'analitzar les dades de la xarxa i determinar si en algun moment hi ha alguna fallada. Qualsevol usuari de la xarxa haurà de poder inscriure's i rebre una notificació en forma de correu electrònic cada cop que es detecti una fallada que l'usuari estigui interessat en tenir-ne informació.

Aquest sistema d'alarma es podria implementar de formes distintes. Una forma seria utilitzant software ja existent i adaptant-lo per a que faci la tasca que volem arribar a aconseguir (com per exemple Nagios [9]) o bé es podria crear un petit i simple programa des de 0, que amb una interfície senzilla sigui capaç de analitzar les dades proporcionades per la pàgina de qMp Wireless Mesh [10] i fer les notificacions corresponents.

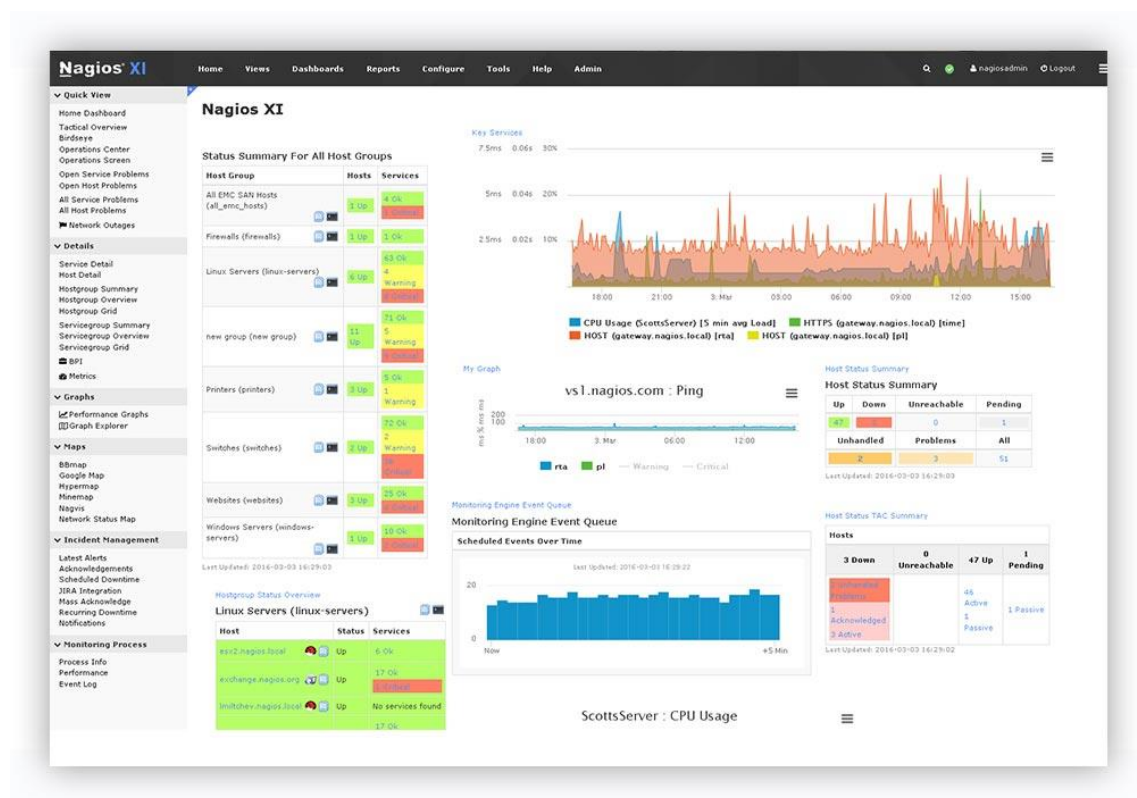


Figura 4 Exemple de la interfície gràfica de Nagios

Es necessitarà una bon coneixement del estat habitual de la xarxa per a poder indentificar amb prou probabilitat les fallades.

4.2. Possibles obstacles

Hi ha diversos factors que poden ser considerats com a possibles obstacles en la realització d'aquest treball.

En primer lloc, al treballar amb eines i programaris que no s'han utilitzat abans, la corba de aprenentatge pot ser difícil i causar un increment en el temps planejat.

Un altre punt que pot portar complicacions és la mateixa estructura de la xarxa, que al ser una xarxa heterogènia, les diferents característiques dels nodes podrien causar falses alarmes, per exemple.

4.2.1. Errors de programació

Com és habitual en tots els projectes de desenvolupament de software, un dels majors obstacles que es poden trobar són els errors a l'hora de programar. Caldrà escriure el codi amb molta cura i anar revisant i comprovant que el funcionament és el correcte i que el codi i les comandes que es van generant no van acumulant errors.

4.2.2. Manca de coneixements

Un altre factor molt important serà la manca de coneixements sobre l'entorn en el que es treballarà i les eines que s'utilitzaran. Caldrà dedicar el temps suficient a entendre i aprendre com funcionen, com a mínim per a poder realitzar el propòsit d'aquest projecte.

4.2.3. Calendari

El temps ajustat que es disposa per realitzar el projecte pot arribar a ser un obstacle, ja que obliga a treballar amb un termini i qualsevol altre obstacle que es trobi podria fer que el temps estimat per a la realització del projecte es vegi incrementat.

5. Planificació temporal

El projecte té una duració d'uns 4 mesos, començant el dia 19 de febrer i acabant el dia 25 de juny. La estimació de la carga de treball és d'unes 450 hores de feina. Aquest quadrimestre, les lectures es realitzen la setmana del 25 de juny fins el dia 29 de juny i, si no hi ha cap desviació important, es realitzarà en aquesta setmana la lectura del projecte.

Es realitzarà per una única persona que s'encarrega de totes les diferents tasques que es descriuen en els apartats següents. Això implica que les tasques s'aniran realitzant de forma seqüencial i les tasques tindran dependències de precedència amb les anteriors, ja que no es començarà una tasca sense que s'hagi acabat la anterior. Tenint en compte això, caldrà seguir la planificació de forma rigorosa ja que el camí serà únic i el resultat del projecte surt de la suma de totes les tasques a realitzar.

5.1. Descripció de les tasques

Per a una millor planificació del projecte, s'ha dividit la realització d'aquest en diverses tasques que s'expliquen a continuació:

5.1.1. Fita inicial

Aquesta fase consisteix en descriure i organitzar el projecte. L'objectiu és deixar definits i especificats mitjançant documentació l'abast, la planificació temporal, els costos i altres informacions necessàries per a dur a terme el projecte.

5.1.2. Estudi de les eines

En aquesta fase s'estudien quines seran les eines més adients per a la realització del projecte. Concretament s'estudiaran quines plataformes de servidors web i bases de dades es faran servir, així com els llenguatges de programació que s'usaran per adequar aquests a les necessitats del projecte.

5.1.3. Implementació

Hi ha tres fases d'implementació que corresponen a les 3 distintes aplicacions necessàries. En totes es realitzarà un aprenentatge per tal de tenir nocions de la eina en qüestió i saber donar-ne un bon us. En primer lloc s'haurà de configurar nagios per a que sigui capaç de analitzar les adreces IP que es registrin (efectuant un ping sobre aquestes i esperant la resposta) i fer la notificació via mail. En segon lloc, implementar la base de dades que guardi la relació dels usuaris registrats amb el seu correu electrònic i amb les adreces que estan interessats en obtenir informació. Finalment caldrà realitzar una interfície web que permeti registrar-se als usuaris, indicant el mail i les adreces en les que tenen interès.

5.1.4. Integració de les aplicacions

En aquesta fase s'ajuntaran les distintes implementacions que s'han realitzat per tal de que funcionin de forma conjunta correctament i es realitzaran les comprovacions adients per tal d'assegurar-ne el correcte funcionament.

5.1.5. Documentació

Consisteix en la redacció de la memòria amb tots els resultats obtinguts i la preparació de la defensa oral davant del tribunal que valorarà el treball realitzat.

5.2. Dependències

En la realització del projecte i, concretament el desenvolupament de les diferents tasques, ve determinat per les tasques que ja hagin estat realitzades. Així doncs, podem dir que algunes de les tasques descrites tenen relacions de dependència i precedència entre elles. En la següent taula es representen aquestes relacions entre tasques:

Tasca	Dependent de
Fita inicial	
Estudi de les eines	Fita inicial
Implementació	Estudi de les eines
Unir les aplicacions	Implementació
Documentació	Totes

Taula 1 Dependències entre tasques

Com podem veure a la taula, un cop acabada la fita inicial es podrà començar amb l'estudi de les eines a utilitzar.

Un cop s'hagin decidit les eines es podrà procedir a la implementació de les 3 aplicacions (que no tenen dependència entre si, es poden realitzar de forma independent).

Seguidament, al haver completat totes les implementacions es podran posar en comú i fer-ne el testeig.

Finalment, pel que fa la documentació, el redactat de la memòria final s'anirà realitzant paral·lelament amb la realització del projecte, així que per redactar la part corresponent a una tasca, es depèn de que aquesta hagi estat realitzada.

5.3. Pla d'acció i valoració d'alternatives

Segons el Racó, els Treballs de Fi de Grau poden tenir les lectures en dos torns, un a l'abril i l'altre al juny. Per motius temporals, la lectura d'aquest projecte es realitzarà en el torn de juny, ja que sinó, no hi hauria prou temps per a realitzar correctament el projecte.

Hi ha un gran nombre de imprevistos que podrien arribar a afectar la data de la entrega. Possiblement, el més crític és que hi hagi algun retràs en la fase d'implementació que faci que tot el projecte vagi endarrerit. És el punt on més problemes poden haver, ja que el desenvolupador haurà de usar i tractar amb tecnologies i problemes que no havia tractat. Algun endarreriment en qualsevol de les tasques provocaria que les estimacions que es realitzen a continuació respecte els temps de dedicació a cadascuna de les tasques canviïn bastant.

Si hi hagués alguna tasca que es compliqués massa, es podria optar per reduir els objectius d'aquella tasca en concret i simplificar-la el necessari per a poder complir amb els límits establerts.

Si per contra, sobrés temps respecte al planejat, es podrien ampliar les funcionalitats de l'aplicació resultant.

5.4. Calendari del projecte

5.4.1. Estimació d'hores per tasca

Totes les tasques que es realitzaran tenen una assignació d'hores aproximada que representa el temps que es tardarà en realitzar-les. Permetrà planificar de manera correcta el temps que el treballador ha de dedicar a cadascuna d'aquestes tasques.

Tasques	Hores
Fita inicial	75
1. Definició de l'abast	20
2. Planificació temporal	10
3. Gestió econòmica i sostenibilitat	10
4. Presentació preliminar	10
5. Plec de condicions	10
6. Presentació oral i document final	15
Estudi de les eines	10
1. Estudi d'eines adients	10
Implementació de Nagios	90
1. Adquisició dels coneixements	10
2. Anàlisi	10
3. Implementació	70
Implementació BD	50
1. Adquisició dels coneixements	10
2. Anàlisi	10
3. Implementació	40
Implementació web	90
1. Adquisició dels coneixements	10
2. Anàlisi	10
3. Implementació	70
Testeig	85
1. Unir les implementacions	75
2. Comprovar el correcte funcionament	10
Documentació	50
1. Redacció de la memòria	40
2. Preparació de la defensa	10
Total	450

Taula 2 Planificació temporal de les tasques

5.4.2. Diagrama de Gantt

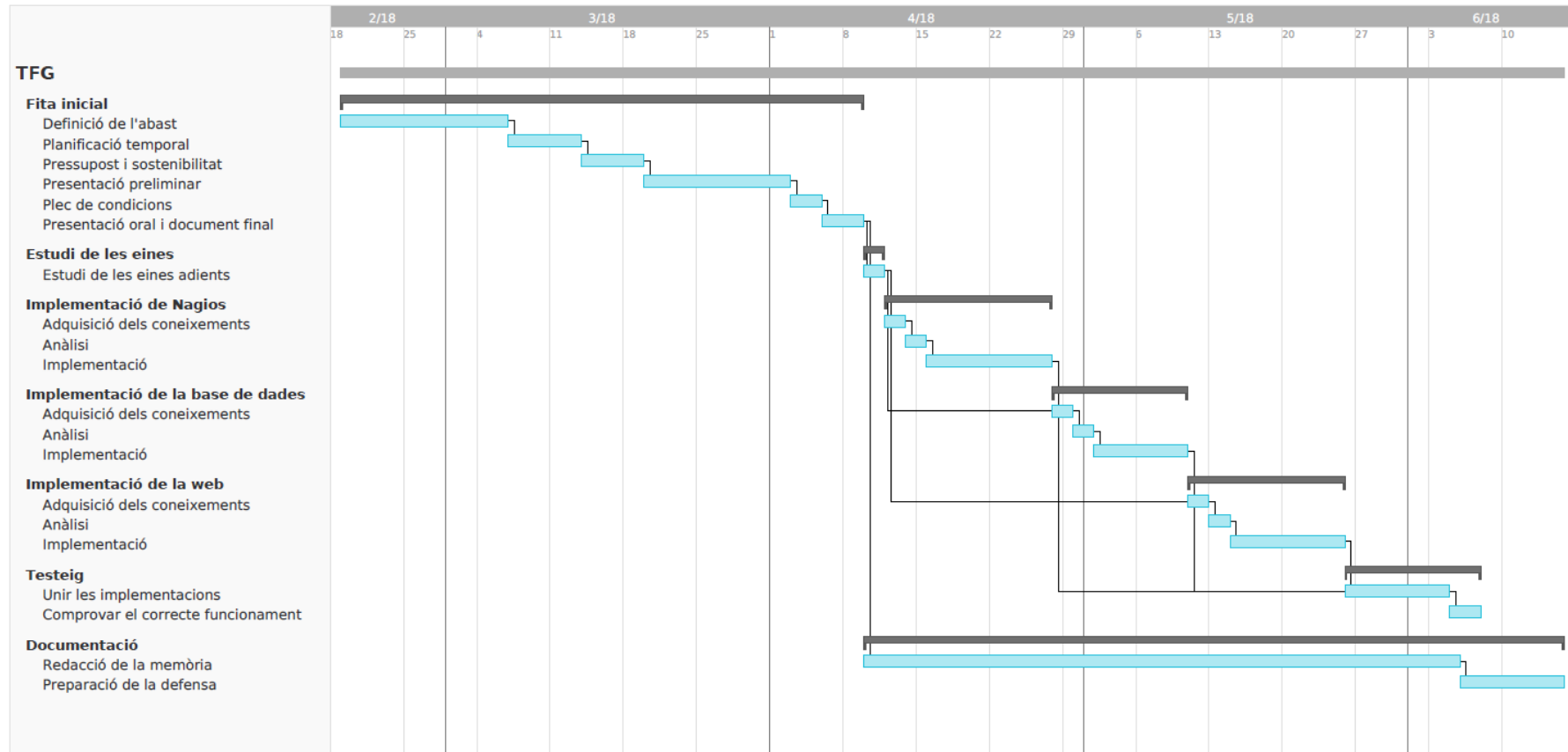


Figura 5 Diagrama de Gantt

5.5. Imprevistos

Tot i que la planificació realitzada sí que s'ha correspost bastant amb la realitat, el fet de que sorgissin alguns imprevistos ha fet que algunes de les tasques tinguessin un temps de realització més elevat.

Per una banda, la implementació de Nagios ha sigut bastant més extensa del que es pensava ja que l'adquisició dels coneixements necessaris per a entendre correctament com configurar-lo de forma correcta ha estat més lenta del esperat, donat que és un programa complex i triar quines són les funcions que es necessitaven per a realitzar el projecte ha sigut un repte. També hi ha hagut el inconvenient que per a poder dur a terme la configuració de Nagios, hi havia la obligació de tenir instal·lat i funcionant un servidor de correu que tingués la capacitat d'enviar correus a qualsevol altre servei de correu, com els que usen els usuaris de la xarxa. Aquesta implementació va requerir el temps d'estudi i aprenentatge i també el de implementació, que va ser més llarg del esperat al produir-se un petit error de programació que va ser difícil de detectar i corregir posteriorment.

Per l'altra, el segon problema que ha afectat la resolució del projecte ha sigut la instal·lació de la base de dades, que degut a un problema en la instal·lació i configuració, va portar més treball del esperat de posar en marxa.

Tasques	Hores
Fita inicial	75
Estudi de les eines	10
Implementació de Nagios	90+30
Implementació BD	50+20
Implementació web	90
Testeig	85+10
Documentació	50
Total	510

Taula 3 Temps dedicat per tasca real

Com es pot comprovar a la taula 3, sí que s'ha hagut de invertir més temps en algunes tasques, però entra dins d'un marge raonable i esperable, ja que aquests retards han vingut donats per imprevistos i per errors que s'han hagut de anar solucionant.

6. Recursos

6.1. Recursos personals

Els únics recursos personals que es disposen per a la realització del projecte serà un treballador, l'autor del projecte. Aquest treballador realitzarà varies funcions per tal de poder finalitzar totes les tasques planejades i poder realitzar amb èxit el projecte.

El temps que es dedicarà al projecte serà d'unes 30 hores setmanals, podent variar segons la carrega de treball que tingui el treballador. També hi haurà setmanes en les que caldrà dedicar més hores per falta de temps per acabar el projecte o bé per poder resoldre qualssevol imprevist que hi hagi per poder complir amb els objectius previstos.

6.2. Recursos materials

Pel que fa als recursos materials, es farà ús de diversos hardwares i softwares que es llisten a continuació.

6.2.1. Hardware

Recurs	Finalitat
Ordinador portàtil HP 250 G6 amb Windows 10 i Ubuntu	<i>Ordinador personal des d'on es treballarà en el projecte.</i>
Ordinador de sobretaula personalitzat amb Windows 10 i Ubuntu	Ordinador personal des d'on es treballarà en el projecte.
Màquina virtual a càrrec del director del projecte	Equip sobre el que s'instal·larà el software necessari i està connectat a la xarxa Guifi·Sants.
Connexió a Internet	Eina necessària per a realitzar la recerca i per a fer funcionar les altres aplicacions.

Taula 4 Recursos Hardware

6.2.2. Software

Recurs	Finalitat
Google Drive	Plataforma on es compartiran fitxers i es realitzaran les còpies de seguretat necessàries.
Nagios	Programa amb el qual es realitzarà la monitorització de l'estat de la xarxa.
Servidor Apache	Servidor web on es posarà en marxa la web per registrar-se al sistema d'alertes.
Base de dades MySQL	Base de dades on es guardarà la informació necessària dels usuaris registrats.
OpenVPN	Eina de connectivitat que permet connectar-se a la màquina virtual de forma remota.
Git	Software de gestió de versions
GitHub	Repositori en línia
Microsoft Office 2010	Eines usades per a crear la documentació i les presentacions.

Taula 5 Recursos Software

7. Gestió econòmica del projecte

Donat que en aquest projecte no s'està produint ningun nombre d'unitats, sinó que estem desenvolupant un servei software que s'oferirà als clients, no es consideraran costos d'activitat relatius al volum de producció. Per tant tots els costos seran considerats com costos d'estructura.

Totes les activitats que s'han mostrat anteriorment, tenen lligats uns costos, tant de recursos humans, hardware, software i d'altres que són indirectes. El pressupost que es calcula deriva de les tasques presentades al diagrama de Gantt. A part, també s'hauran de considerar les despeses de contingència i imprevistos. Per cadascun dels costos s'especificarà si es tracta de un cost directe o indirecte.

7.1. Estimació dels costos

7.1.1. Costos de recursos humans

Aquest tipus de costos són els que van lligats als salaris dels treballadors i el temps que estaran desenvolupant el projecte. Com que aquest projecte es realitza per una única persona, haurà de realitzar totes les funcions, cap de projecte, analista, dissenyador, desenvolupador i enginyer QA.

Rol	Hores estimades	Salari (€/h)	Cost estimat
Cap de projecte	125	20	2500
Analista	25	15	375
Dissenyador	15	15	225
Desenvolupador	275	12	3300
Enginyer QA	10	15	150
Total	450		6550

Taula 6 Cost de recursos humans

Tasques	Hores	Recurs	Salari	Cost
Fita inicial	75			1500
Definició de l'abast	20	Cap	20	400
Planificació temporal	10	Cap	20	200
Gestió econòmica i sostenibilitat	10	Cap	20	200
Presentació preliminar	10	Cap	20	200
Plec de condicions	10	Cap	20	200
Presentació oral i document final	15	Cap	20	300
Estudi de les eines	10			150
Estudi d'eines adients	10	Analista	15	150

Implementació de Nagios	90			1110
Adquisició dels coneixements	10	Desenvolupador	12	120
Anàlisi i disseny	10	Analista + Dissenyador	15	150
Implementació	70	Desenvolupador	12	840
Implementació BD	50			630
Adquisició dels coneixements	10	Desenvolupador	12	120
Anàlisi i disseny	10	Analista + Dissenyador	15	150
Implementació	30	Desenvolupador	12	360
Implementació web	90			1110
Adquisició dels coneixements	10	Desenvolupador	12	120
Anàlisi i disseny	10	Analista + Dissenyador	15	150
Implementació	70	Desenvolupador	12	840
Testeig	85			1050
Unir les implementacions	75	Desenvolupador	12	900
Comprovar el correcte funcionament	10	Enginyer QA	15	150
Documentació	50			1000
Redacció de la memòria	40	Cap	20	800
Preparació de la defensa	10	Cap	20	200
Total	450			6550

Taula 7 Cost de recursos humans segons activitats

La primera taula que mostrem (taula 1), conté el cost directe total per a cada rol del treballador i a la següent (Taula 2), té desglossats els costos per activitat en comptes de per rol. Aquestes taules es generen a partir del calendari previst i amb el diagrama de Gantt i ens permeten tenir una visió detallada de a on va parar el diner que s'inverteix en recursos humans, en quines activitats i en quins moments.

Tots els salaris indicats s'han estimat fent ús de les pàgines de *Michael Page* i *Page Personnel*.

7.1.2. Costos de recursos materials

En aquest apartat s'inclouen totes les despeses de recursos materials de hardware i software que s'utilitzen per a la realització del projecte. Aquests costos es consideren indirectes, ja que no depenen del projecte que s'està desenvolupant, sinó que son despeses que es tindrien de independentment de si s'està desenvolupant o no.

Pel que fa al hardware, trobem el material físic que s'utilitzarà per a dur a terme el programa.

Producte	Preu (€)	Unitats	Vida útil (anys)	Amortització/h	Hores	Amortització
Ordinador portàtil HP 250 G6	500€	1	4	0.071*€/h	112.5	8 €
Ordinador de sobretaula personalitzat	1200€	1	4	0.17*€/h	337.5	5757€
Màquina virtual	120€	1	0.5	0,027€/h	4380	120€
Total						185€

Taula 8 Costos indirectes per recursos materials hardware

Per a la taula de costos de recursos materials hardware s'ha tingut en compte per a la amortització que un any té uns 220 dies laborables i que cada dia té 8 hores laborables. Per a la màquina virtual es considera que funciona les 24 hores del dia. També s'ha considerat que, aproximadament, s'utilitzarà el ordinador portàtil un 25% del temps de realització del projecte, mentre que la resta s'efectuarà al de sobretaula.

Pel que fa el software, no tindrem en compte els seus costos ja que tot el que es fa servir o bé és Open Source, o tenim disponibles versions de prova suficientment llargues com per a dur a terme el projecte.

7.1.3. Costos generals

En aquesta categoria entren tots aquells costos que no es poden classificar com a recursos humans ni materials. En son un exemple el preu del transport que s'utilitzi, el cost del lloc on es realitzi el projecte, etc.

Producte	Preu (€/mes)	Cost estimat (€)
Connexió a Internet	50 €	200€
T-Jove 5 zones	94 €	376€
Total		576€

Taula 9 Costos generals

7.1.4. Imprevistos

En tots els projectes es fa una reserva de diners per tal de poder cobrir la despesa addicional que causaria algun problema imprevist durant el projecte que faci que el cost previst es vegi elevat. En aquest projecte, la partida per a imprevistos que es reservarà serà d'un 15% dels costos tant de recursos humans, materials i generals.

En cas de que no hi hagi cap imprevist, aquesta despesa no s'utilitzarà.

Producte	Percentatge	Preu	Cost
Recursos humans	15 %	6550€	982€
Recursos materials	15 %	177.65€	27€

Recursos generals	15 %	576€	87€
Total		7303.65€	1096€

Taula 10 Costos d'imprevistos

7.1.5. Contingència

Tot i que ja hem tingut en compte que poden sorgir imprevistos, els quals no hem reconegut, hi ha certs esdeveniments que podem intuir que es realitzaran i per tant, podem incloure un pla de contingència per poder seguir amb el curs del projecte.

Risc acceptat	% Ocurrència	Cost estimat conseqüències	Exposició al risc
Avaria de l'ordinador	5%	500€	25€
Retràs en la implementació (15 dies)	15%	75h * 12€/h = 900€	135€
Total			160€

Taula 11 Costos de contingència

7.1.6. Pressupost final

Per acabar, es posen en comú tots els càlculs efectuats en els apartats anteriors i es proporciona una última taula que conté el pressupost final per a realitzar el projecte.

Concepte	Cost
Recursos humans	6550€
Recursos materials	178€
Recursos generals	576€
Imprevistos	1096€
Contingència	160€
Total	8560€

Taula 12 Pressupost final

Aquest pressupost ha estat suficient per a cobrir el cost real del projecte, ja que amb les reserves d'imprevistos i de contingència ja n'hi ha hagut prou per a poder cobrir les despeses generades per les hores extra que s'han hagut d'invertir al treball.

7.2. Control de gestió

En aquest projecte hi ha pocs elements que es puguin comprar, ja que els recursos materials s'han comprat fa un temps i les despeses generals existirien igualment tot i no

estar realitzant el projecte. Així doncs, l'únic que es pot controlar són els costos dels recursos humans.

Per a realitzar el control, es realitzarà una comprovació al final de cada fase del projecte que consistirà en comparar les hores reals dedicades a una tasca i les hores teòriques assignades. En un cas ideal, les hores reals i les estimades haurien de ser bastant pròximes i, en la mesura possible, intentar que les primeres no superin per una diferència gran a les segones.

Si resulta que al concloure el projecte les hores reals dedicades superen les que s'havien estimat, s'haurà de fer ús dels fons de contingència i d'imprevistos per cobrir les despeses addicionals.

Per veure les desviacions que s'hauran produït durant la execució del projecte es faran servir les següents fórmules:

- Desviacions en la realització de tasques (en cost):
 $(\text{cost estimat} - \text{cost real}) * \text{consum hores real}$
- Desviacions totals en la realització de tasques:
 $(\text{cost estimat total} - \text{cost real total})$
- Desviacions de recursos humans:
 $(\text{cost estimat} - \text{cost real}) * \text{consum hores real}$
- Desviacions d'un recurs *hardware* (en cost):
 $(\text{cost estimat} - \text{consum real}) * \text{cost real}$
- Desviacions totals de recursos:
 $(\text{cost estimat total} - \text{cost real total})$

8. Metodologia i rigor

8.1. Fita inicial

8.1.1. Mètode de treball

Per al desenvolupament del projecte es farà una metodologia en cascada. Aquesta metodologia consisteix en un procés seqüencial per el disseny del software. Consta de 5 fases, en les que es defineix el problema i els requeriments que s'han de tractar; s'analitza la millor forma de donar una solució i es dissenya; s'implementa el disseny que s'havia acceptat; es prova que el sistema funciona de forma correcta i finalment es realitza el manteniment del producte final. Es pot veure d'una forma més clara i esquemàtica a la següent figura:

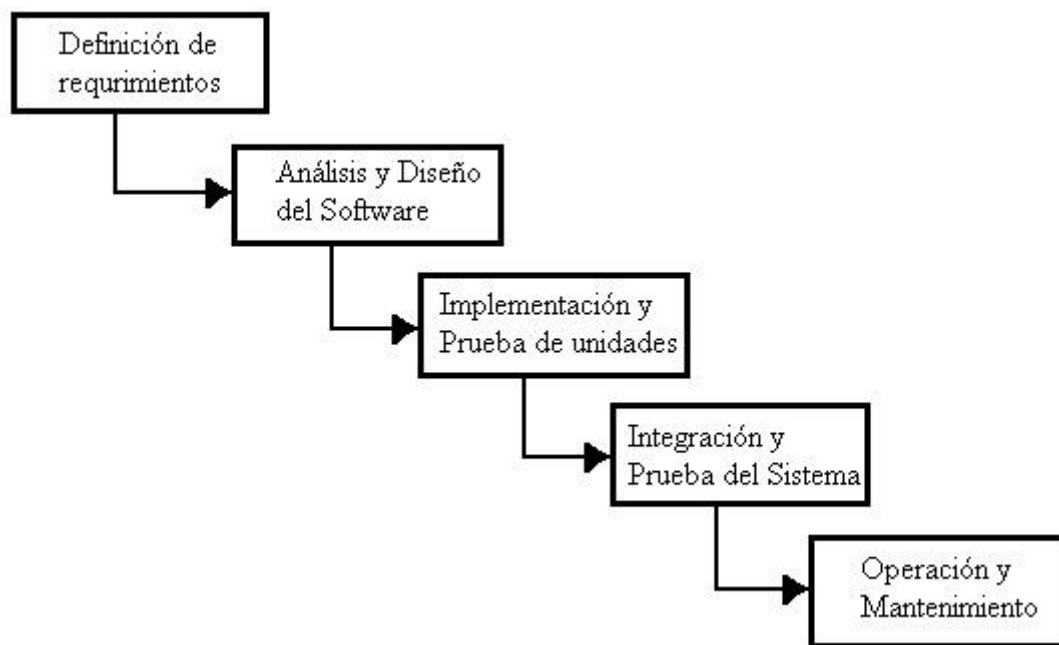


Figura 6 Graf representatiu de la metodologia cascada

8.1.2. Eines

S'utilitzaran varies eines per a la realització del projecte.

S'ha de decidir si serà més eficient utilitzar algun software ja existent per a realitzar la monitorització de les dades de la xarxa WMN (com podria ser una versió de Nagios [9])

preparada especialment per a satisfer els requeriments del projecte) o bé implementar un senzill software específic que aprofiti les dades que ja es recullen de forma periòdica en la web del qMp Wireless Mesh, ja que es poden extreure de forma senzilla en format JSON.

Per a tenir un control de les versions s'utilitzarà Git [11] i GitHub [12] per a poder tenir un repositori que es pugui accedir online.

També s'utilitzarà Google Drive [13] per a tenir els arxius necessaris al núvol i fer-ne còpies de seguretat que també seran accessibles online.

La primera d'elles és openvpn, una eina que permet connectar una màquina a una VPN. Aquesta aplicació és essencial en la realització del projecte ja que la implementació es realitza en una màquina remota (tal i com s'explica al apartat següent) i per a connectar-s'hi, es necessita tenir accés a la seva VPN.

Com a aplicació de monitorització de la xarxa, finalment s'ha decidit usar Nagios Core, que és una aplicació que permet de forma relativament senzilla, configurar tot un seguit de mètriques i serveis que es van avaluant periòdicament i té la capacitat d'enviar notifikacions directament al detectar-se un error en algun d'aquests. La configuració d'aquesta aplicació comporta una gran part del treball i es podria dir que és la part on més esforç i dedicació s'ha dedicat.

També era essencial la presència d'un servidor de correu, ja que el canal escollit per enviar les notifikacions d'error als usuaris és precisament el correu electrònic. Com a servidor de correu s'ha escollit Postfix, ja que és un servidor ràpid, fàcil d'administrar i segur.

Com a mètode de guardar les dades que es van produint a mesura que es van registrant els usuaris, també era essencial l'ús d'algun tipus de base de dades. S'ha escollit un servidor de MySQL, ja que és un llenguatge que ja he estudiat en alguna assignatura de la carrera i, per tant, no resultaria massa complicat d'entendre i configurar.

El projecte també necessitava una interfície on els usuaris poguessin registrar-se per a poder seleccionar els nodes que estan interessats en tenir informació, així que s'ha decidit posar en marxa una simple pàgina web en un servidor d'Apache2.

Finalment, per tal de poder tenir un control de versions i tenir còpies de seguretat dels fitxers i el codi que s'ha anat generant s'ha fet ús de Git conjuntament amb Github i també del servei d'emmagatzemament al núvol de Google, el Google Drive.

8.1.3. Entorn de treball

L'entorn de treball sobre el qual s'ha treballat es tracta de una màquina virtual. Aquesta màquina virtual ha estat posada en marxa per el director del projecte, Llorenç Cerdà i té la particularitat que forma part de la xarxa mesh de Guifi-Sants.

Per a poder connectar-se a aquesta màquina virtual s'ha proporcionat els fitxers necessaris per a realitzar una connexió VPN, que dona accés remot a la xarxa i mitjançant una terminal de comandes, a través de ssh, es pot accedir amb un usuari i la respectiva clau a la màquina virtual.

En aquesta màquina virtual s'hi ha realitzat tota la configuració necessària, és a dir, s'hi ha instal·lat totes les aplicacions esmentades a l'apartat anterior i s'han configurat i posat en marxa de forma que la aplicació que s'ha dissenyat funcioni.

A part del treball realitzat dins de aquesta màquina virtual, també s'ha fet treball fora, ja que a l'hora de programar els scripts necessaris, resultava més senzill i còmode fer-ho amb una interfície gràfica i amb més opcions i espai de treball.

8.1.4. Mètode de validació

Es realitzaran reunions amb el director que serviran per comprovar que el progrés del projecte és l'adequat. La validació del software que s'implementa es provarà per a testejar que detecta els errors, que és capaç d'enviar les notificacions que pertoquen i finalment, fer ús del programa en el cas real, és a dir, posar-lo en funcionament i que els usuaris finals rebin els correus de notificació de errors.

9. Desenvolupament i implementació

En aquest capítol s'explicarà tot el procés de implementació que s'ha seguit per tal de dur a terme el projecte. S'aportarà el coneixement necessari per entendre els passos realitzats, des de la instal·lació fins a la configuració final que s'ha realitzat. Per a que hi hagi una major claredat en la explicació, aquest capítol es divideix en seccions que es corresponen a les tasques realitzades, tal i com es poden veure a l'apartat 5, que correspon a la planificació temporal.

9.1. Nagios

Un cop estudiades i decidides les aplicacions a usar, el primer pas que es va realitzar va ser instal·lar i posar en marxa el servidor de Nagios en la màquina virtual. Aquest va ser el primer pas que es va decidir realitzar ja que és el que més feina portaria a la hora de aprendre i entendre el seu funcionament.

A continuació es realitza una breu explicació sobre què és Nagios i les funcions que aporta.

Nagios, conegut actualment com a Nagios Core, és una aplicació software lliure i de codi obert que permet la monitorització de sistemes, xarxes i infraestructures. Ofereix monitorització i serveis d'alerta per servidors, commutadors, aplicacions i serveis.

Nagios té moltes utilitats entre les quals s'inclouen:

Monitorització comprensiva

Permet controlar diferents sistemes, aplicacions, serveis, sistemes operatius, protocols de xarxes, etc. amb una única aplicació. Mitjançant APIs i scripts fa que sigui altament personalitzable.

Correcció d'errors

Mitjançant les notificacions, permet comunicar els errors de forma ràpida i els controladors d'events permeten el reinici automàtic de aplicacions i serveis fallits.

Informes

Els informes de disponibilitat asseguren el compliment dels SLA.

Hi ha disponibles històrics que proporcionen un registre d'alertes, notificacions, interrupcions i respostes d'alertes.

Arquitectura extensible

La integració de diferents aplicacions tant internes com de tercers és molt senzilla gràcies a múltiples API disponibles.

Hi ha cents de complements o plugins desenvolupats per la comunitat que estenen la funcionalitat bàsica de Nagios.

Comunitat

Nagios té una comunitat de més d'un milió d'usuaris en tot el món, que desenvolupa un gran nombre de complements que ajuden a ampliar les funcionalitats principals de Nagios.

Visibilitat i Consciència

Ofereix una vista centralitzada de tota la infraestructura de TI.

Hi ha una interfície web que permet una visualització detallada dels estats.

9.1.1. Ús en el projecte

Dins del projecte que s'està realitzant, l'ús del Nagios és essencial, ja que és la aplicació amb la qual realitzem les comprovacions de l'estat dels nodes que hi ha a la xarxa. També s'aprofitarà el sistema de notificacions que inclou el software per a realitzar les alertes que s'envien als usuaris.

9.1.2. Consideracions tècniques

Per a poder entendre correctament la implementació, personalització i posada en marxa del programa, cal tenir certs coneixements de com funciona i com es pot configurar correctament el Nagios.

L'ús més bàsic que es pot donar del Nagios és configurar els serveis o *services*. Els serveis permeten la programació de comprovacions periòdiques, les quals executen certes comandes i en detecten el seu estat i, si escau, generen la notificació que pertoca. Aquesta serà la configuració que s'usa en aquest projecte.

Un cop s'instal·la Nagios en el sistema i es posa en marxa, i havent configurat correctament el servidor d'Apache2 que utilitza per a la seva interfície web, podem veure-la:

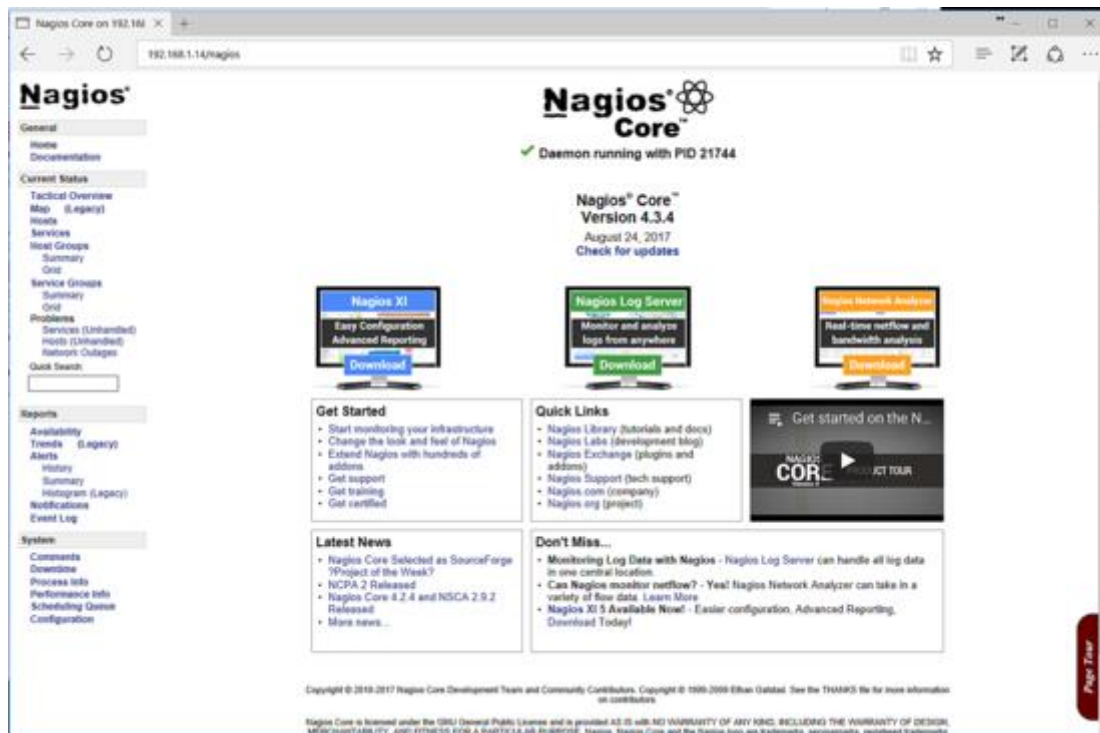


Figura 7 Interfície web de Nagios

Mitjançant aquesta web tenim accés ràpid i visual a totes les opcions que ens aporta Nagios i podem, entre d'altres, comprovar l'estat dels seus serveis actius com en el següent exemple:

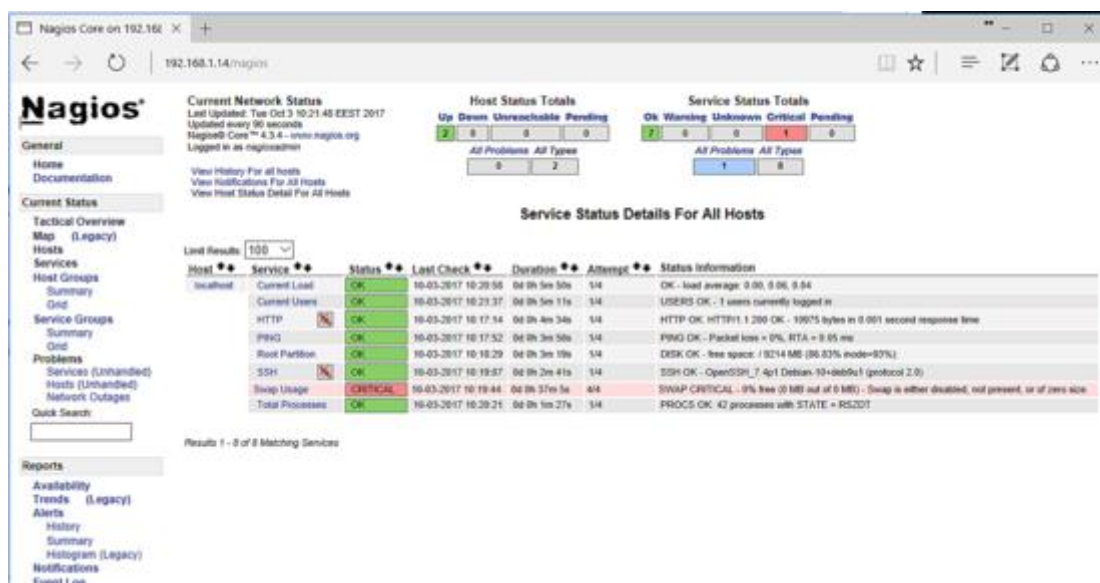


Figura 8 Vista dels serveis de Nagios en la interfície web

Nagios, si no s'especifica, s'instal·la a la ruta de `/usr/local/nagios`, en aquest directori trobem tot el que necessita nagios per a funcionar i per a poder configurar-lo. Per aquest projecte, els fitxers que tenen una rellevància més gran són els que hi ha inclosos dins del directori `/usr/local/nagios/etc/objects`.

En concret, els arxius que s'han modificat són els següents:

localhost.cfg

A localhost hi ha una configuració senzilla per a poder monitoritzar màquina local. En aquest fitxer trobem varies definicions com ara:

-Host

El host s'usa per a definir un servidor físic, dispositiu, *Workstation*, etc. que hi ha a la xarxa. Una definició d'exemple seria:

```
define host{
    host_name          bogus-router
    alias              Bogus Router #1
    address            192.168.1.254
    parents            server-backbone
    check_command       check-host-alive
    check_interval     5
    retry_interval     1
    max_check_attempts 5
    check_period       24x7
    process_perf_data   0
    retain_nonstatus_information 0
    contact_groups     router-admins
    notification_interval 30
    notification_period 24x7
    notification_options d,u,r
}
```

-Hostgroup

De forma opcional es pot definir també un hostgroup, que és un grup de tots els hosts que en formen part. Per exemple:

```
define hostgroup{
    hostgroup_name     novell-servers
    alias              Novell Servers
    members             netware1,netware2,netware3,netware4
}
```

-Service

La definició d'un servei s'utilitza per identificar un "servei" que corre en el host. El terme servei és ambigu, ja que es pot tractar d'un servei real, com ara POO, SMTP, HTTP o algun altre tipus de mètrica associada a un host, com ara un ping, el nombre d'usuaris connectats o l'espai lliure al disc. Els serveis permeten la execució periòdica de

comprovacions, especificades per l'atribut *check_command*. Si aquesta comprovació resulta amb una alerta, es genera i s'envia la corresponent notificació. Per exemple:

```
define service{
    host_name                linux-server
    service_description      check-disk-sda1
    check_command             check-disk!/dev/sda1
    max_check_attempts       5
    check_interval           5
    retry_interval           3
    check_period              24x7
    notification_interval    30
    notification_period      24x7
    notification_options     w,c,r
    contact_groups            linux-admins
}
```

En aquest exemple s'està definint un servei que té com a host *Linux-server* i la seva funció és comprovar el disc que correspon al sda-1. Amb un interval de 5 minuts s'avalua la comanda *check-disk!/dev/sda1*. Si el resultat genera notificació, s'enviarà als usuaris que formin part del contact_group Linux-admins, amb un temps de 30 minuts entre cada alerta.

contacts.cfg

Aquest fitxer de configuració conté la informació de tots els contactes i grups de contactes que hi ha en el sistema de monitorització. Aquests contactes es poden referenciar en el host i en les definicions de serveis, per tal de que siguin avisats per les notificacions relacionades amb aquests serveis.

Troblem dos tipus de definicions:

-Contact

La definició d'un contacte fa referència a un usuari que s'ha de contactar quan hi hagi un esdeveniment o un problema a la xarxa. Un exemple de contacte és:

```
define contact{
    contact_name              jdoe
    alias                     John Doe
    host_notifications_enabled 1
    service_notifications_enabled 1
    service_notification_period 24x7
    host_notification_period 24x7
    service_notification_options w,u,c,r
    host_notification_options d,u,r
    service_notification_commands notify-by-email
    host_notification_commands host-notify-by-email
    email                     jdoe@localhost.localdomain
    pager                     555-
5555@pagergateway.localhost.localdomain
}
```

```

address1          xxxxx.xyyy@icq.com
address2          555-555-5555
can_submit_commands 1
}

```

D'aquest contacte, podem dir, entre altres, que té com a nom d'usuari *jdoe* i el seu al·lies és John Doe. Té activades les notificacions i li arriben en qualsevol dia i hora de la setmana. Per a les notificacions farà servir les comandes de *notify-by-email* i *host-notify-by-email*.

-Contactgroup

Un *contactgroup* o grup de contactes és un grup d'un o més usuaris agrupats per tal de facilitar el enviar alertes i notificacions. Si a un servei s'indica un *contactgroup* en el seu camp *contact_groups*, s'enviaran les notificacions d'alerta a tots els usuaris que en formin part.

```

define contactgroup{
    contactgroup_name      novell-admins
    alias                  Novell Administrators
    members                jdoe,rtobert,tzach
}

```

Aquest *contactgroup* referenciat com *novell-admins* inclou a tres usuaris: *jdoe*, *rtobert* i *tzach*.

commands.cfg

Finalment, l'últim fitxer que s'ha modificat és el de *commands.cfg*. En aquest fitxer tenim definides les comandes que s'utilitzen des de les definicions dels host, serveis i contactes.

-commands

Els *commands* o comandes són això, definicions de comandes. Les comandes que es poden definir inclouen comprovacions de serveis, notificacions de serveis, controladors d'events, comprovacions de host, etc. Les comandes poden contenir macros, però s'ha d'assegurar que són vàlides per a tots els moments en que es poden executar.

Mitjançant els *commands* es poden definir les comprovacions necessàries en qualsevol infraestructura i determinar els paràmetres d'error per tal de generar les notificacions.

```

define command{
    command_name    check_pop
    command_line    /usr/local/nagios/libexec/check_pop      -H
$HOSTADDRESS$
}

```

En aquest exemple s'ha definit la comanda *check_pop*, que executa el fitxer executable que està en l'adreça especificada amb els paràmetres *-H* i l'adreça del host.

9.1.3. Instal·lació

Un cop fet el breu resum de les configuracions que s'han usat per a poder aconseguir que Nagios funcioni per a resoldre el problema que es planteja en aquest projecte, es passa a explicar el procés de instal·lació de Nagios Core en l'entorn de treball i la seva primera configuració.

El primer pas va ser comprovar que les dependències que necessita estan presents en el sistema, que es pot realitzar amb la següent comanda:

```
apt install autoconf gcc libc6 make apache2-utils libgd-dev
```

Un cop totes les dependències i els paquets necessaris per compilar Nagios al nostre sistema Debian, s'ha de descarregar de la web oficial la versió que es desitgi, en aquest cas es va triar la versió 4.3.4, que era la més nova i actualitzada en el moment de realitzar la instal·lació.

```
wget  
https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.3.4.tar.gz
```

Un cop descarregat, cal descomprimir el fitxer i realitzar la configuració corresponent mitjançant l'executable *configure* que es troba dins dels fitxers. Un cop realitzada la configuració ja es pot compilar mitjançant *make all*, tal i com podem veure a les imatges següents:

```

config.status: creating t-tap/Makefile
config.status: creating include/config.h
config.status: creating lib/snprintf.h
config.status: creating lib/iobroker.h

Creating sample config files in sample-config/ ...

*** Configuration summary for nagios 4.3.4 2017-08-24 ***:

General Options:
-----
Nagios executable: nagios
Nagios user/group: nagios,nagios
Command user/group: nagios,nagios
Event Broker: yes
Install ${prefix}: /usr/local/nagios
Install ${includedir}: /usr/local/nagios/include/nagios
Lock file: /run/nagios.lock
Check result directory: ${prefix}/var/spool/checkresults
Init directory: /etc/init.d
Apache conf.d directory: /etc/apache2/sites-enabled
Mail program: /bin/mail
Host OS: linux-gnu
IOBroker Method: epoll

Web Interface Options:
-----
HTML URL: http://localhost/nagios/
CGI URL: http://localhost/nagios/cgi-bin/
Traceroute (used by WAP): /usr/sbin/traceroute

Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.

root@nagios:~/nagios-4.3.4# make all

```

Figura 9 Sortida de terminal de la configuració

```

make install-exfoliation
- This installs the Exfoliation theme for the Nagios
web interface

make install-classicui
- This installs the classic theme for the Nagios
web interface

*** Support Notes *****
If you have questions about configuring or running Nagios,
please make sure that you:

- Look at the sample config files
- Read the documentation on the Nagios Library at:
https://library.nagios.com

before you post a question to one of the mailing lists.
Also make sure to include pertinent information that could
help others help you. This might include:

- What version of Nagios you are using
- What version of the plugins you are using
- Relevant snippets from your config files
- Relevant error messages from the Nagios log file

For more information on obtaining support for Nagios, visit:

https://support.nagios.com

*****
Enjoy.

root@nagios:~/nagios-4.3.4# _

```

Figura 10 Sortida de terminal al realitzar la compilació de Nagios

Acte seguit era necessari crear l'usuari de sistema i el seu grup i donar-li els permisos necessaris per accedir als recursos web de Apache per tal de poder mostrar la interfície web:

```
useradd nagios
usermod -a -G nagios www-data
```

Un cop fet, només queda instal·lar l'aplicació i posar-la en marxa:

```
make install-init
systemctl enable nagios.service
make install-config
```

Un cop instal·lat, cal també instal·lar també els fitxers de configuració per Apache, executant la següent comanda:

```
make install-webconf
```

També s'ha creat un usuari anomenat nagiosadmin, que serà de moment el que es podrà connectar a la interfície web. Per donar els permisos executem:

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users
nagiosadmin
```

Finalment, queda reiniciar el servidor d'apache per a poder carregar la nova configuració i també arrancar Nagios:

```
systemctl restart apache2
systemctl start nagios
```

Amb aquesta configuració realitzada, Nagios ja està en funcionament en el sistema i ja es pot veure la interfície web si s'accedeix amb el usuari nagiosadmin que s'ha creat.

Tot i ja tenir corrents Nagios, no és del tot operatiu encara, ja que no hi ha cap servei de control configurat, cosa que es podria comprovar si s'entra en l'apartat de serveis actius que hi ha disponible a la interfície web, com es veu en la següent imatge:



Figura 11 Pestanya de serveis de Nagios sense cap servei actiu

Així doncs, vaig buscar alguns *plugins* bàsics per realitzar aquesta monitorització i els vaig instal·lar.

El primer pas, és un altre cop instal·lar totes les dependències necessàries, que són les següents:

```
apt install libmcrypt-dev make libssl-dev bc gawk dc build-essential snmp libnet-snmp-perl gettext libldap2-dev smbclient fping default-libmysqlclient-dev
```

Els *plugins* que s'han instal·lat formen part del repositori de Github anomenat Nagios Plugins i per a aconseguir-los, s'ha introduït la següent comanda:

```
wget https://github.com/nagios-plugins/nagios-plugins/archive/release-2.2.1.tar.gz
```

Un cop descarregat, cal descomprimir, configurar, compilar i instal·lar, tal i com s'ha fet anteriorment per a instal·lar Nagios i posteriorment reiniciar el servei de Nagios per a que aquests canvis tinguin efecte.

Així ja s'han creat diversos fitxers de configuració senzills de mostra que es poden veure a la web mostrada anteriorment:

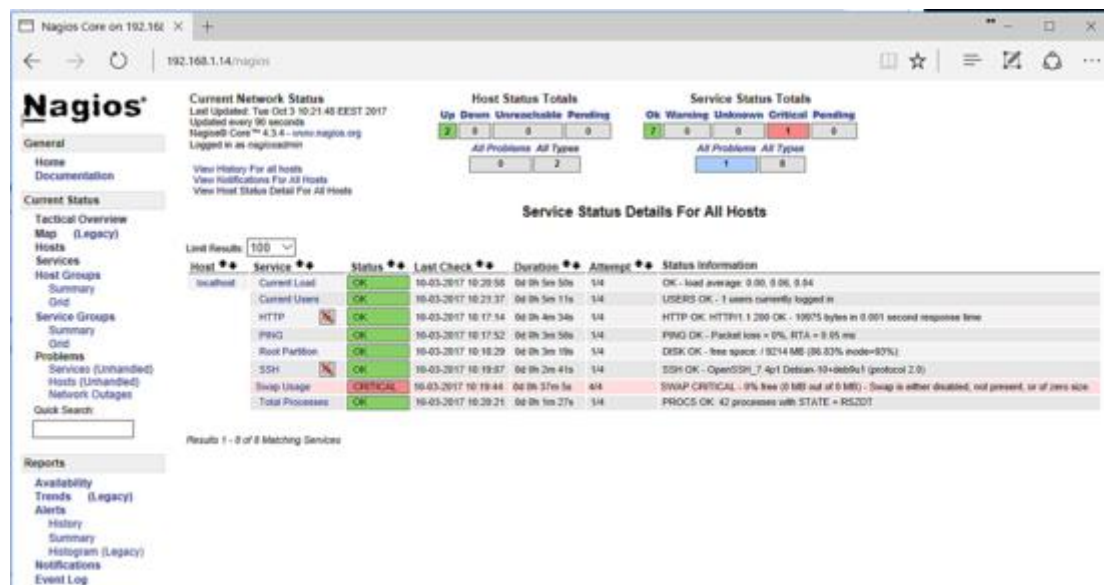


Figura 12 Pestanya de serveis de Nagios amb serveis d'exemple actius

En aquest cas, veiem que ja s'han creat distints serveis de comprovació i es mostra el seu estat actual, ja sigui correcte, de avís o crític.

En aquest punt ja tenim instal·lat el servidor de Nagios, juntament amb tot un conjunt de *plugins* que aporten funcions molt diverses i útils que podrem configurar. Tot i això, per a poder acabar de realitzar la configuració de Nagios es necessiten les dades dels usuaris per poder tenir els contactes corresponents i també són necessàries les dades dels nodes, és a dir, els noms dels nodes i les seves direccions IP, per a poder programar els serveis que avaluaran el seu estat i generaran les notificacions corresponents.

Aquesta configuració final s'explica en el capítol 9.4., juntament amb la integració de tots els serveis i implementacions realitzades.

9.1.4. Servidor de correu

Tot i no ser una configuració que correspongui directament amb Nagios, es va trobar que era imprescindible tenir un servidor de correu instal·lat en l'entorn de treball, ja que sense aquest, seria impossible enviar els correus de les notificacions. Aquest fet no s'havia contemplat en la planificació inicial i va comportar un cert retràs en la realització de les altres tasques que hi havia planejades, per tant, es van dedicar més hores setmanals al projecte.

En aquest capítol s'explica el procés i els problemes que hi va haver a la hora de instal·lar i configurar el servidor de correu.

Com a servidor de correu, es va decidir que el més indicat seria usar Postfix, ja que és un servidor senzill, ràpid i fàcil de configurar. També es fa ús del servidor d'smtp de google per a poder arribar a qualsevol servidor de correu que puguin fer servir els potencials usuaris de la xarxa.

Un cop decidit el software a utilitzar, el primer pas és efectuar la instal·lació. Amb la següent comanda es pot instal·lar el servidor conjuntament amb totes les utilitats que són necessàries per a poder dur a terme la configuració per a Nagios.

```
apt-get install postfix mailutils libsasl2-2 ca-certificates  
libsasl2-modules
```

Un cop instal·lades totes les aplicacions necessàries, passem a editar el fitxer de configuració de postfix per a poder usar el servidor de smtp de google. El fitxer que s'ha de modificar en aquest cas es troba a */etc/postfix/main.cf* i modificar-lo amb els següents paràmetres:

```
relayhost                =[smtp.gmail.com]:587  
smtp_sasl_auth_enable    =                               yes  
smtp_sasl_password_maps  =  hash:/etc/postfix/sasl_passwd  
smtp_sasl_security_options =                             noanonymous  
smtp_tls_CAfile          =                               /etc/postfix/cacert.pem  
smtp_use_tls = yes
```

També s'ha de crear el fitxer que es menciona al camp *smtp_sasl_password_maps* que conté la informació de la compta de google que s'usa per a enviar els correus. Per a fer-ho, creem el fitxer a la ruta mencionada i afegim:

```
[smtp.gmail.com]:587 correugmail@gmail.com:contrasenya
```

Establim els permisos d'aquest fitxer i actualitzem la configuració de Postfix

```
chmod 600 /etc/postfix/sasl_passwd  
postmap /etc/postfix/sasl_passwd  
chmod 600 /etc/postfix/sasl_passwd.db
```

L'últim pas és comprovar si els certificats que tenim instal·lats són correctes mitjançant la comanda:

```
cat /etc/ssl/certs/Thawte_Premium_Server_CA.pem | tee -a  
/etc/postfix/cacert.pem
```

Donat que en el meu cas no va sortir cap error, no va ser necessària cap configuració addicional. Finalment es pot refrescar el servei de postfix per a que els canvis realitzats tinguin efecte.

```
service postfix reload
```

Un cop realitzada la configuració, el nostre sistema ja hauria de ser capaç d'enviar correus electrònics a través del compte de google que hem indicat a postfix. Per a provar-ho, es pot enviar un simple correu mitjançant la terminal de comandes per a verificar que la configuració ha estat correcta.

```
echo "Prova Postfix + nagios" | mail -s "notificacio nagios"
correu@servidor.com
```

Tot i que aquesta configuració és bastant senzilla, degut a un petit error humà al escriure malament un paràmetre, va acabar resultant bastant complicat, però un cop trobat el error, el sistema funcionava correctament.

Per a que Nagios sigui capaç d'enviar correus, només cal configurar, dins del fitxer de comandes, una comanda que s'encarregui d'enviar els correus amb el format que s'indiqui. En el cas d'aquest projecte, la comanda que s'ha usat és la següent:

```
#notify-service-by-email' command definition
define command{
    command_name      notify-service-by-email
    command_line       /usr/bin/printf "%b" "***** Aquest correu
s'ha generat automaticament al detectar-se un problema amb
un          node.          *****\n\nNotification          Type:
$NOTIFICATIONTYPE$\n\nService:      $SERVICEDESC$\nDate/Time:
$LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$\n" |
/usr/bin/mailx -s "** Detectat un problema amb un node de la
xarxa Guifi Sants **" $CONTACTEMAIL$
}
```

Per a que s'utilitzi aquesta configuració com a mètode de notificació, en la definició de cada contacte dins del fitxer de configuració de Nagios de `contacts.cfg`, s'haurà d'especificar en el camp de `service_notification_commands`.

Amb aquesta configuració realitzada, ja es pot programar un servei per comprovar que el nostre mètode per enviar correus funciona correctament. En aquest cas, es va programar un servei anomenat `emailcheck`, el qual intenta comprovar amb pings una adreça IP que no es troba present a la xarxa, de forma que cada cop que es fa la comprovació doni com a resultat error i per tant, hagi de crear la notificació. Aquest és el servei programat:

```
define service{
    use                               local-service
    host_name                         localhost
```

```
service_description      emailcheck
check_command            emailcheck
contact_groups           prova
}
```

En aquest exemple, quan es detecta el error, envia la notificació a tots els contactes que formen part del grup “prova”, dins el qual hi ha un usuari propi que té configurades les notificacions amb la comanda que s’ha explicat anteriorment. Al saltar el error el correu que es rep és el següent.

**** Detectat un problema amb un node de la xarxa Guifi Sants ****



nagiossergi@gmail.com <nagiossergi@gmail.com>

19:59

Para: sergipica@hotmail.com

***** Aquest correu s'ha generat automaticament al detectar-se un problema amb un node. *****

Notification Type: PROBLEM

Service: emailcheck

Date/Time: Sun Jun 24 17:59:11 UTC 2018

Additional Info:

CRITICAL - Host Unreachable (10.1.26.98)

Figura 13 Correu generat per emailcheck

9.2. Implementació de interfície web

El servidor triat per realitzar la configuració de la interfície web finalment ha estat Apache, ja que és el servidor més popular actualment. Està molt ben documentat i s'ha usat àmpliament durant la història de la web, així que era una bona opció per defecte per a muntar un lloc web.

En aquest capítol s'explica tot el procediment realitzat per a instal·lar i posar en marxa la interfície web amb la que els usuaris interactuen per tal de introduir el seu correu en el que volen rebre les notificacions així com marcar els nodes sobre els que volen rebre els avisos en cas d'algun error.

Així doncs, s'explica el procediment de la instal·lació, la senzilla web realitzada i finalment els scripts usats per a poder fer funcionar correctament el sistema.

9.2.1. Instal·lació

La instal·lació de Apache2 és realment molt senzilla i ràpida de realitzar. Per dur-la a terme s'han seguit els següents passos.

En primer lloc és bo assegurar-se de que el sistema està correctament actualitzat i té les últimes versions de software instal·lades. Així doncs, executem:

```
apt-get update && apt-get upgrade
```

Un cop comprovades les actualitzacions i instal·lades si cal, només s'ha d'instal·lar el servidor mitjançant la comanda:

```
apt-get install apache2
```

Amb això fet, podem comprovar si la instal·lació ha estat correcte d'una forma molt senzilla i gràfica, ja que apache té una web de mostra en la qual s'indica que el servidor està funcionant. La web en qüestió es pot veure a la següent figura:

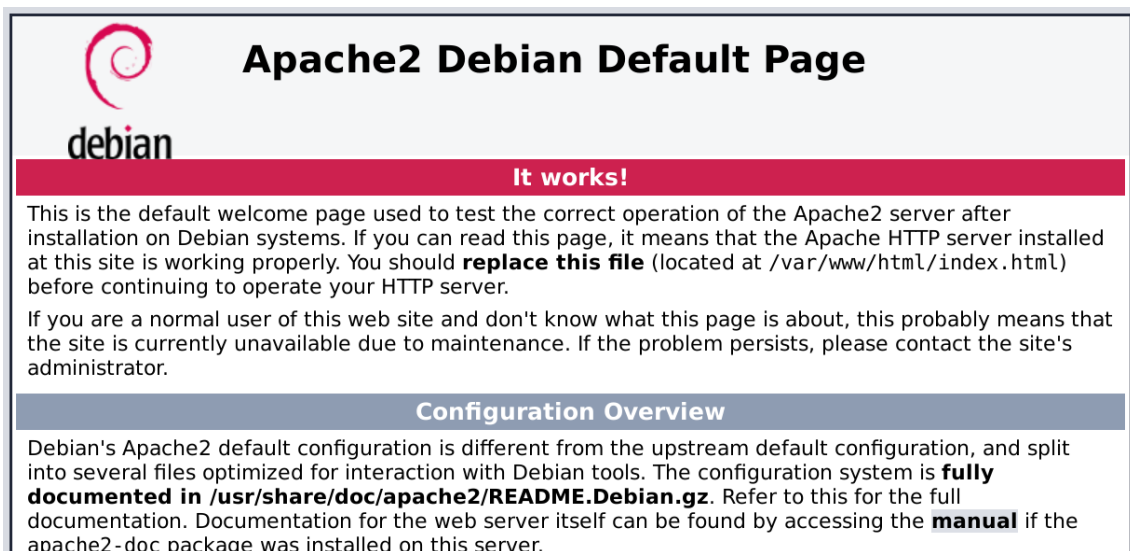


Figura 14 Pàgina per defecte de apache

En aquesta web podem trobar una senzilla explicació de com usar apache i indica el directori on s'han de modificar o afegir les distintes webs que es poden accedir.

En el cas d'aquest projecte, la web o interfície que s'ha programat és una senzilla web en html, en forma de formulari, que permet al usuari introduir les dades d'una forma senzilla i intuïtiva.

9.2.2. Web

Com s'ha explicat anteriorment, la web que s'ha realitzat és un senzill formulari escrit en html i, per a poder recopilar les dades que s'envien, es fa servir un script cgi escrit en python, que recull les dades amb el mètode de POST.

Pel que fa la pàgina web (inclosa en els annexos d'aquesta memòria), es tracta d'un formulari amb un camp que permet escriure un correu electrònic, el qual serà l'identificador de l'usuari i també el correu al qual s'enviaran les notificacions i tot un seguit de *checkboxes*, que són unes caselles que es poden marcar i cada casella correspon a un node de la xarxa.

Per a poder crear aquesta web, era necessari tenir les dades de tots els nodes que hi ha a la xarxa. Les dades finalment s'han extret de la pàgina de visualització de l'estat de la xarxa <http://dsg.ac.upc.edu/qmpsu/>, la qual proporciona la utilitat de descarregar totes les dades que es mostren visualment en format JSON.

Per tant, el primer pas era extreure la informació que seria útil per a la realització del projecte d'aquest JSON. Els camps que s'han decidit útils han estat el nom del node, per tal de poder identificar de forma senzilla quines dades corresponen a cada node i per a

poder proporcionar als usuaris també una forma de seleccionar els que vulguin; i les direccions IP de cada node.

Donat que hi ha nodes que tenen més d'una direcció Ipv4 assignada, es va haver d'escollir-ne només una, i el criteri per a seleccionar-la és de escollir la direcció que correspongui a la direcció del node dins de la xarxa interna de Guifi-Sants, ja que així s'assegura que el servidor de Nagios sigui capaç de determinar si es troben disponibles o no.

Així doncs, per extreure aquestes dades, donat el gran nombre de nodes presents, la millor solució va ser crear un petit script que fes la feina per nosaltres.

El script escrit en javascript utilitzat es pot trobar en els annexos i el que fa és mostrar una pàgina web amb una llista en la que es mostren els noms dels nodes seguits de les seves direccions ip.

Aquestes dades s'han usat amb un altre script per a crear totes les entrades necessàries per a crear la web en el format correcte.

Finalment, l'únic pas que quedava realitzar era administrar d'alguna forma les dades que es generen amb aquesta web. Per a fer-ho, s'ha creat un cgi escrit en python també.

9.2.3. CGI

El CGI (*Common Gateway Interface*) o interfície d'entrada comú és una tecnologia molt important i molt usada en les webs. Permet a un client, en aquest cas el navegador web, sol·licitar dades d'un programa que s'executa al costat del servidor web. Així doncs, és un mecanisme molt útil per a poder transferir dades entre el client i el programa. Les CGI permeten aportar moltes funcionalitats a les webs i van ser les primeres pràctiques que van permetre tenir continguts dinàmics.

De forma resumida, les CGI són scripts que s'executen al servidor quan un client ho demana, administrant les dades que li arriben i produint una sortida per al client.

En aquest projecte, el propòsit del script CGI que es realitza és bàsicament aconseguir recopilar les dades que introdueix l'usuari i poder-les guardar. Com que per a poder guardar les dades és necessari l'ús de la base de dades, aquesta part s'explicarà en l'apartat d'integració mentre que aquí, s'explica el procediment que s'ha de realitzar amb python per a poder obtenir les dades enviades de la web.

En els annexes es pot trobar el script sencer de cgi que s'ha usat, tot i que en aquest apartat es comenten les parts més importants per el propòsit de recopilar les dades.

Recopilar les dades amb el llenguatge python és un procediment molt senzill ja que hi ha una llibreria que ho facilita i que per tant, cal importar al nostre script:

```
import cgi
```

Seguidament cal crear l'objecte que conté les dades que s'han enviat. En el cas del script que s'ha generat per el projecte s'ha anomenat form:

```
form = cgi.FieldStorage()
```

A partir d'aquest objecte ja es poden extreure tots els resultats que s'han rebut fent ús de la funció `getvalue()`. Uns exemples són:

```
email = form.getvalue('email')
```

```
form.getvalue('BCCanBruixa20Rd6')
```

Amb `get value`, s'obté el valor que es proporciona de la web que correspon amb el camp que es passa com a paràmetre de la funció.

Per determinar quins són els nodes en els que l'usuari està interessat, es crea una llista amb els valors de tots els camps corresponents als nodes i després es seleccionen els que han estat marcats i es col·loquen en una altra llista que correspon a les opcions triades.

Els camps que s'han de guardar posteriorment són el email i la llista de nodes.

9.3. Implementació de base de dades

L'última utilitat que s'ha hagut de instal·lar i configurar per a dur a terme aquest projecte és la de la base de dades. S'ha decidit usar el servidor de MYSQL de base de dades donat que és un de molt conegut i del que hi ha molta documentació i informació. A part, ja havia usat aquesta base de dades en altres assignatures cursades, com ara a PTI (Projecte de Tecnologies de la Informació).

La base de dades s'haurà de configurar per tal de que sigui capaç de guardar la informació sobre tots els nodes que s'extreu dels JSON provinents de la web del qMP. També s'haurà d'encarregar de guardar les relacions entre usuaris i els nodes sobre els quals estan interessats.

9.3.1. Instal·lació

En aquest cas la instal·lació també es força senzilla de realitzar, tot i que en l'entorn de treball es van trobar certs problemes, els quals es van poder solucionar. Tot el procés seguit s'explica a continuació.

El primer pas, com és habitual, consisteix en comprovar si existeixen actualitzacions disponibles per el nostre sistema i si és el cas, instal·lar-les. Acte seguit ja es pot procedir a instal·lar el servidor de MYSQL.

```
sudo apt-get install mysql-server
```

Segons la informació trobada, durant la instal·lació s'hauria d'haver demanat certs paràmetres de configuració com ara la contrasenya que es vol triar per al usuari root. Per un error en el sistema de Debian aquesta configuració no es va poder realitzar i per tant, per a poder accedir al servidor es va haver de configurar manualment la contrasenya per accedir a la base de dades.

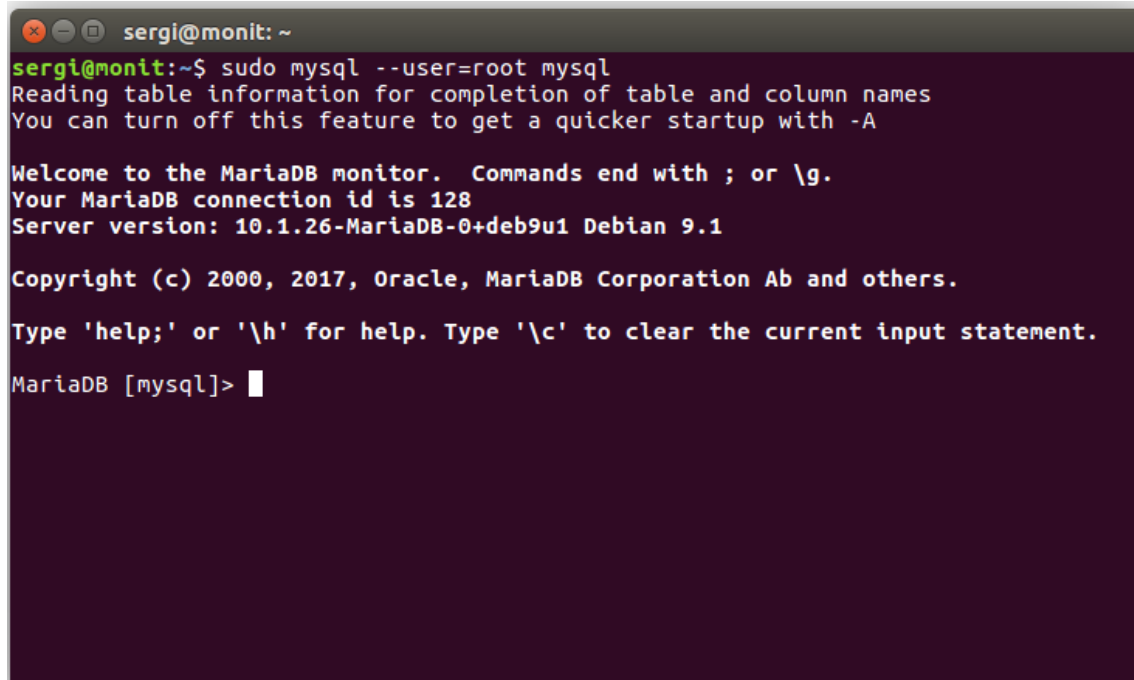
El procediment a realitzar va ser engegar el servidor sense carregar les taules de grant, el qual permet accedir a la base de dades sense haver de introduir cap contrasenya.

Un cop parat el servidor de mysql es van tenir que introduir les següents comandes per desactivar aquests *flags*:

```
sudo mkdir /var/run/mysqld
sudo chown mysql /var/run/mysqld
sudo mysqld_safe --skip-grant-tables&
```

En aquest punt ja es pot accedir a la base de dades mitjançant la comanda:

```
sudo mysql --user=root mysql
```



```
sergi@monit: ~
sergi@monit:~$ sudo mysql --user=root mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 128
Server version: 10.1.26-MariaDB-0+deb9u1 Debian 9.1

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [mysql]> 
```

Figura 15 Interfície de la base de dades

Un cop dins, ens apareix el quadre de diàleg del servidor (tal i com veiem a la figura 15) i per a efectuar el canvi de contrasenya s'ha de introduir:

```
update user set authentication_string=PASSWORD('nova-contrassenya') where user='root';
flush privileges;
```

Un cop ja tenim la contrasenya per l'usuari root, només cal reiniciar el servei corresponent al mysql per a tornar a establir les *grant-tables*. No cal efectuar ningun pas més per acabar la instal·lació, així que només quedava crear i configurar les taules.

9.3.2. Configuració

Com a configuració, cal crear les taules necessàries per a guardar la informació i introduir la que ja tenim disponible i que no es modificarà.

La primera decisió presa és que per a guardar les dades es crearien dues taules. La primera taula serveix per tenir una relació entre cada nom de node i la seva direcció ip corresponent. La segona, és també una taula amb dues columnes i relaciona els usuaris amb els noms dels nodes que han seleccionat.

Per a crear les taules primer es necessita una instància de base de dades. Per a crear-la s'executa, un cop hem entrat a mysql amb l'usuari que hem configurat:

```
create database testdb;
```

Per a usar aquesta *data base* creada escrivim:

```
Use testdb;
```

Ara, un cop dins, ja podem crear i modificar les taules que creem.

```
CREATE TABLE NodeIP ( NomNode VARCHAR (50) NULL , IP VARCHAR (50) NULL );
```

```
CREATE TABLE Usuaris ( Usuari VARCHAR (50) NULL, NomNode VARCHAR (50) NULL );
```

Aquestes dues bases de dades són suficients per complir el propòsit del projecte i només cal omplir-les. La taula de nodes la podem tenir completa a partir de les dades que s'extreuen del JSON, però la taula dels usuaris s'haurà de crear dinàmicament a mesura que els usuaris es vagin registrant, és per això que en aquest apartat s'explica només com s'omple la primera taula, mentre que la segona s'explica en el següent apartat corresponent a la integració.

Donat que existeixen molts nodes, el més còmode era crear un script que tingués la capacitat d'anar emplenant la base de dades de forma automàtica. El llenguatge escollit un altre cop és el python, ja que hi ha una llibreria que facilita l'accés a bases de dades de mysql directament en des de l'script.

Per a instal·lar els mòduls necessaris per a poder fer servir MySQL directament a python hem de, simplement, executar la comanda:

```
sudo apt-get install python-pip python-dev libmysqlclient-dev
```

Un cop instal·lades les llibreries necessàries, cal escriure el script.

Es necessita connectar a la base de dades mitjançant un usuari i també es crea un cursor, que és a partir de on s'executaran les sentències.

```
db = MySQLdb.connect(host="localhost",      # your host
                      user="root",          # username
                      db="prova")          # name of the database

cur = db.cursor()
```

Un cop connectat amb la base de dades, aprofitem les dades que havíem extret anteriorment per a crear la pàgina web (noms dels nodes seguits de les seves direccions ip). Obrim l'arxiu que les conté i el separe per línies, de forma que es pot accedir com si es tractés d'una llista de python, tenint en compte que de forma alternada aparexien els noms i les ip, tal que:

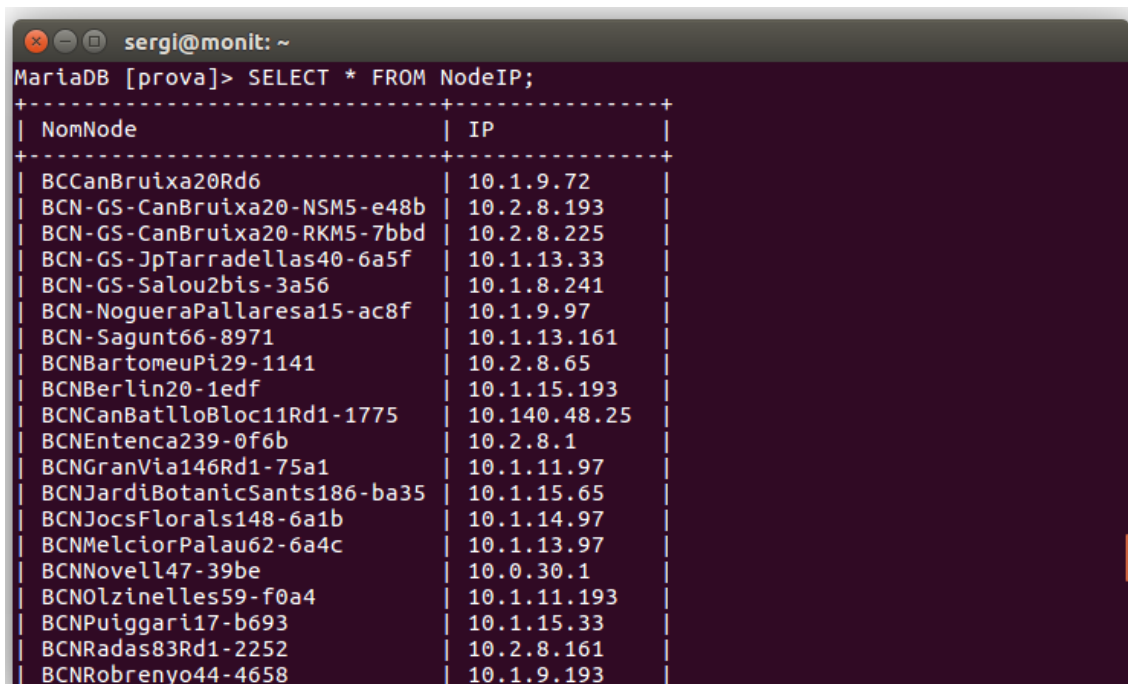
```
Dades[x] #conté nom
Dades[x+1] #conté ip
```

Així amb un bucle for podem recórrer totes les entrades, crear les sentències sql (en forma de string) i executar-les usant el cursor que hem creat:

```
for x in lines:
    if (i % 2 == 0):
        aux = "INSERT INTO NodeIp VALUES ('" + x + "', '"
+ lines[i+1] + "');"
        print aux
        cur.execute(aux)
    i = i + 1
```

Amb la comanda *execute* del cursor, s'executa a la base de dades la comanda representada per les instruccions que hem introduït en el string aux.

Un cop executat aquest script, es pot comprovar que la taula ha quedat emplenada correctament si realitzem la consulta corresponent, tal i com es pot veure en la figura següent:



The screenshot shows a terminal window with the prompt 'sergi@monit: ~'. The user has entered the command 'MariaDB [prova]> SELECT * FROM NodeIP;'. The output is a table with two columns: 'NomNode' and 'IP'. The table contains 20 rows of data, each representing a node and its IP address.

NomNode	IP
BCCanBruixa20Rd6	10.1.9.72
BCN-GS-CanBruixa20-NSM5-e48b	10.2.8.193
BCN-GS-CanBruixa20-RKM5-7bbd	10.2.8.225
BCN-GS-JpTarradellas40-6a5f	10.1.13.33
BCN-GS-Salou2bis-3a56	10.1.8.241
BCN-NogueraPallaresa15-ac8f	10.1.9.97
BCN-Sagunt66-8971	10.1.13.161
BCNBartomeuPi29-1141	10.2.8.65
BCNBerlin20-1edf	10.1.15.193
BCNCanBatlloBloc11Rd1-1775	10.140.48.25
BCNEntenca239-0f6b	10.2.8.1
BCNGranVia146Rd1-75a1	10.1.11.97
BCNJardiBotanicSants186-ba35	10.1.15.65
BCNJocsFlorals148-6a1b	10.1.14.97
BCNMelciorPalau62-6a4c	10.1.13.97
BCNNovell47-39be	10.0.30.1
BCNOLzinelles59-f0a4	10.1.11.193
BCNPuiggari17-b693	10.1.15.33
BCNRadas83Rd1-2252	10.2.8.161
BCNRobrenyo44-4658	10.1.9.193

Figura 16 Taula NodeIP plena

9.4. Integració de les tecnologies

Per acabar, un cop instal·lades i configurades totes les aplicacions, l'últim pas a realitzar és acabar de unir-les per a donar el servei final que es tenia com a objectiu. Per a aconseguir-ho, s'haurà de connectar la web amb la base de dades, de manera que les dades que es recullen es puguin guardar de forma persistent i també s'haurà de connectar la base de dades amb la configuració de nagios, per tal de tenir els contactes i els contactgroups correctes per a tots els serveis que controlen l'estat dels nodes.

9.4.1. Connexió web – base de dades

Aquest és l'apartat més senzill de realitzar de la integració i per a dur-lo a terme, s'introdueix al script de cgi de la pàgina web una secció de codi que s'encarrega de guardar de forma correcta les dades a la base de dades.

Es farà ús del mòdul explicat anteriorment al capítol 9.3.2., que permet manipular la base de dades amb una connexió que es realitza directament des del script.

Per a simplificar la implementació, s'ha decidit que si un usuari que ja existeix en el sistema decideix tornar a enviar les seves dades a través del web, es tindran com a nodes que li interessin únicament els que hagi seleccionat en aquesta última selecció, independentment de si els tenia marcats o no anteriorment.

Per assegurar que això es compleix, en primer lloc i mitjançant el cursor, creem la sentència en SQL que permet trobar totes les entrades en les que hi apareix l'usuari i s'esborren.

```
sql = "SELECT * FROM Usuaris WHERE Usuari ="
sql = sql + email + "';"
cur.execute(sql)

if cur.rowcount != 0:
    sql = "DELETE FROM Usuaris WHERE Usuari ="
    sql = sql + email + "';"
    cur.execute(sql)

db.commit()
```

En aquest codi es pot comprovar que primer es busca si el usuari ja existeix a la base de dades i en cas de ser cert, *rowcount* retorna un valor superior a 1 (retorna el nombre de files que s'han generat amb la anterior consulta). Si el usuari està en la base de dades, es declara la sentència sql que el permet esborrar i s'executa. Finalment s'ha de realitzar un commit per tal de que aquest canvi tingui efecte.

La segona tasca que ha de realitzar el script és generar les noves entrades de la base de dades, així com també mostrar una pantalla a l'usuari per a confirmar que la seva sol·licitud s'ha realitzat amb èxit.

```
imprimir = "<html><head><title>enviat  
correctament</title></head><body><p> el email es:  
"+email+"</p>"

i = 0
base = "INSERT INTO Usuaris VALUES ('"
mig = "','"
final = "');"
count = 0
for x in list:
    if not (x is None):
        count = count + 1
        sql = base + email + mig + names[i] + final
        cur.execute(sql)
        db.commit()
        imprimir = imprimir + "<p> Has seleccionat  
el node " + names[i] + "</p>"
        i = i+1
if count == 0:
```

```
imprimir = "<p> No has triat cap node, per tant ja  
no rebràs cap notificació més d'aquest sistema"  
print imprimir
```

En aquest altre fragment de codi (el codi sencer es pot trobar al annex), és on es realitza la inserció del usuari amb tots els nodes que ha seleccionat i es crea la sortida de la interfície web.

Per a cada valor de *list* (llista amb tots els valors dels *checkbox*) que no és nul, es crea la sentència d'inserció a la base de dades i s'imprimeix a la web un missatge que indica que s'ha seleccionat el node. En cas de que no s'hagi seleccionat cap node, el usuari ja no consta a la base de dades, per tant ja no rebrà cap correu de notificació més, tal i com se indicaria a la web que li apareixeria.

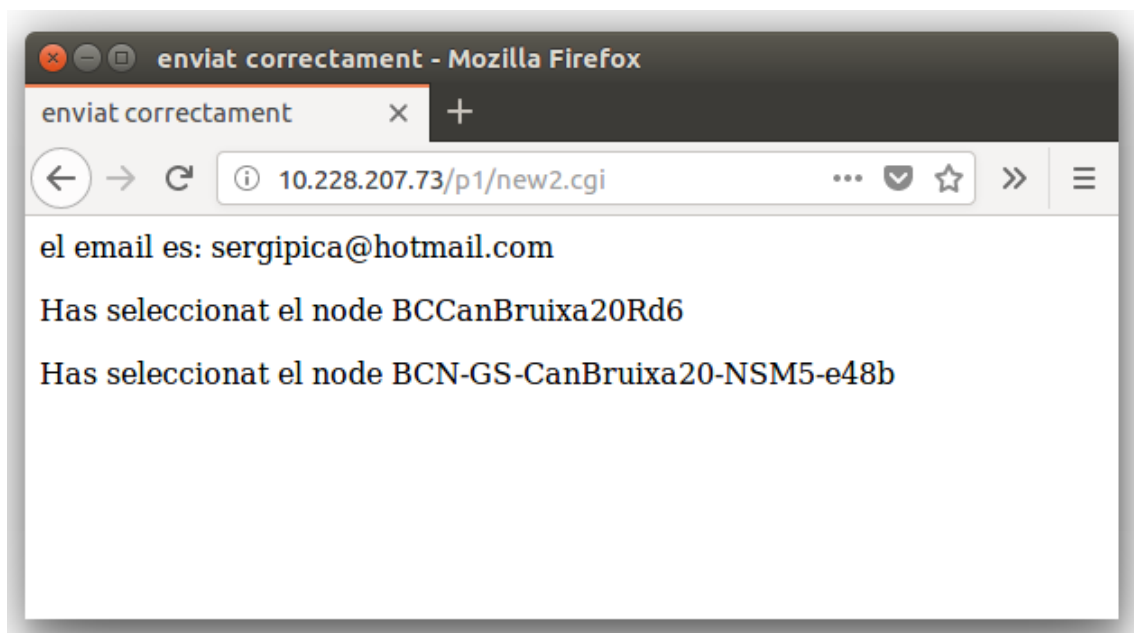


Figura 17 Exemple de sortida al registrar-se

Com es pot veure en la figura 17, un usuari s'ha registrat en el sistema i ha seleccionat dos nodes.

9.4.2. Connexió base de dades – Nagios

Per a acabar de configurar Nagios, s'han de crear diversos objectes de configuració en els diferents fitxers que s'han descrit anteriorment. En concret, s'han de crear els serveis, un per a cada node, que s'encarregaran de anar comprovant el estat dels nodes de forma periòdica; les comandes que fan servir els serveis per a efectuar les configuracions; els contactes que corresponen als usuaris i finalment els contactgroups, un per a cada node/servei, que permetrà agrupar tots els usuaris que estiguin interessats en un mateix node.

D'aquestes quatre tasques que cal fer, les dues primeres seran estàtiques i per tant, només caldrà realitzar un cop, ja que es treballarà sempre amb el mateixos nodes. Les referents als contactes en canvi, hauran de ser dinàmiques i cada cop que es produeixi un canvi en el sistema, s'hauran de ajustar.

Per a crear els serveis, ens tornem a fer ús d'un script en python conjuntament amb el fitxer on tenim els noms dels nodes amb les seves direccions ip que s'ha obtingut a partir del JSON. S'han de generar serveis com el que es mostra a continuació:

```
define service {  
    use local-service  
    host_name localhost  
    check_command BCCanBruixa20Rd6_check  
    contact_groups BCCanBruixa20Rd6  
}
```

Aquest servei usa el template anomenat local-service i es defineix amb el `check_command` *nom_de_node_check* que serà el nom que donarem a les comandes que generem per a cada servei. També s'indica que el `contact_group` al que pertany correspon amb el nom del node.

El script, obre el arxiu i el llegeix dividint per línies, tal com feia en altres exemples. Un cop llegit, recorre la llista de respostes que s'ha generat i va creant un string que conté la plantilla anterior, substituint el nom del node per cada node que es troba i ho imprimeix.

La sortida del script la he redireccionat en un fitxer i fàcilment es poden copiar i afegir al fitxer de objectes de nagios anomenat localhost.cfg.

El procediment per a crear les comandes és bastant semblant, ja que s'haurà de recórrer la mateixa taula i generar les comandes tal i com s'ha fet per els serveis. En aquest cas una comanda d'exemple seria:

```
define command {  
    command_name BCCanBruixa20Rd6_check  
    command_line $USER1$/check_ping -H 10.1.9.72 -w 3000,80% -  
c 5000,99% -p 5  
}
```

El nom de la comanda es defineix com a *node_check* i la comanda a executar correspon al programa `check_ping`, que s'encarrega de fer ping a la direcció ip que se li proporciona i s'especifiquen també els límits dins dels que es considera un warning o un estat crític.

Aquestes comandes s'afegeixen al fitxer `commands.cfg`

Per els altres objectes que s'han de crear, els contactes i els *contactgroups* formen part del mateix fitxer de configuració, el `contacts.cfg`.

S'haurà de crear un contacte per a cada usuari present en la base de dades en el moment de canviar la configuració i també modificar els *contactgroups* per tal de que hi hagi un grup per a cada node i en la llista de membres de cada grup hi figurin els usuaris que l'han triat.

Aquesta configuració es realitza en el moment que un usuari es registra, ja que és el moment en el que es produeixen els canvis en la base de dades així que s'aprofita el mateix script per a realitzar les tasques de configuració, assegurant que un cop finalitza, la base de dades i la configuració de Nagios s'han realitzat amb èxit.

Com que els fitxer `contacts.cfg` no s'ha de escriure de nou cada cop, ja que hi ha alguns usuaris i grups d'usuaris que són fixos i no han de variar, s'ha creat un fitxer que conté les definicions estàtiques. El script el llegeix i els contactes que es creen de forma dinàmica s'hi afegeixen a sota.

En primer lloc explicaré els *contactgroups*. Aquests es defineixen un cop per node i el seu nom coincideix amb el del node. N'és un exemple:

```
define contactgroup {
contactgroup_name BCCanBruixa20Rd6
members sergipica@hotmail.com,
}
```

Aquest grup és el que correspon al node de BCCanBruixa20Rd6 i actualment l'únic usuari que n'està interessat és el del correu que apareix en els membres. En cas d'haver més d'un membre, es van declarant separats per una coma.

```
cur.execute("SELECT      NomNode      FROM      NodeIP")
items      =      cur.fetchall()

for      row      in      items:
    sql = "SELECT Usuari FROM Usuaris WHERE NomNode='" +
row[0]      +      "'"
    cur.execute(sql)
    items2      =      cur.fetchall()
    res = res + "define contactgroup { \ncontactgroup_name
"      +      row[0]      +      "\nmembers      "
    for      x      in      items2:
        res      =      res      +      x[0]      +      ",      "
        res      =      res      +      "\n}\n"
```

Aquest fragment de codi és el encarregat de crear els contactgroups. Per funcionar, el primer que fa és llegir tots els noms de nodes que hi ha a les bases de dades i, per cadascun, busca els usuaris que el tenen marcat i els afegeix als membres del grup.

Quan ha recorregut tots els nodes, *res* conté un string on hi ha tota la configuració acumulada i l'escriu en el fitxer de configuració de Nagios.

De forma semblant, es creen els usuaris, que segueixen el següent model:

```
define contact{
    contact_name sergipica@hotmail.com
    email sergipica@hotmail.com
    service_notification_options    w,u,c,r,f,s
    service_notification_commands   notify-service-by-email
    host_notification_options       d,u,r,f,s
    host_notification_commands      notify-host-by-email
}
```

Aquesta definició de contacte crea el contacte amb nom i correu iguals al correu introduït i es configuren les opcions de notificació per tal de rebre les alertes quan els serveis corresponents ho requereixin.

Per a crear-los en l'script es realitza de forma semblant als contactgroups, de forma que primer es realitza una consulta per obtenir tots els usuaris diferents que hi ha presents a la base de dades i per a cadascun d'aquests usuaris es crea la seva entrada de contacte amb la mateixa configuració de notificacions.

10. Sostenibilitat i compromís social

10.1. Autoavaluació

Actualment, crec que tinc un bon coneixement sobre la sostenibilitat i l'impacte que poden tenir les TIC. La gran majoria dels coneixements que he anat obtenint prové de competències transversals que s'han anat treballant en diferents assignatures, sobretot aquelles que formen part del departament d'arquitectura i, en especial, en les assignatures de APC (Arquitectura del PC), ASMI (Aspectes Socials i Mediambientals de la Informàtica) i CPD (Centres de Processament de Dades).

Aquestes competències m'han ajudat a entendre que tots els projectes o activitats que es realitzen en el entorn de les TIC, tenen un impacte ambiental, social i econòmic que s'ha de tenir molt en compte a la hora de desenvolupar-los.

Si bé tinc la consciència del impacte dels projectes TIC, la major dificultat que tinc seria la de mesurar l'impacte d'aquests mitjançant els indicadors, ja que possiblement és el que menys es treballa en aquestes competències transversals proposades en el pla d'estudi.

Per acabar, crec que és molt bo per a tots els estudiants que es donin a conèixer aquests aspectes sobre les tecnologies que probablement serien ignorats o si més no, no es tindrien presents. M'ha permès donar-me compte que amb petites accions es pot arribar a un canvi bastant significatiu i crec que en el futur podré aplicar amb satisfacció totes les tècniques per millorar els projectes en els que formi part, ja sigui en temes socials, econòmics i ambientals com també aspectes de justícia social, equitat, diversitat, transparència i ètica, entre d'altres.

10.2. Dimensió ambiental

Actualment, en la xarxa de Guifi-Sants, quan hi ha un error, no hi ha una forma immediata de detectar quin és el node que ha deixat de funcionar i, per tant, els responsables d'aquesta xarxa (que són els propis usuaris) han de trobar el problema i solucionar-lo.

Amb la solució proposada, aquests errors es podran detectar d'una forma molt més fàcil i ràpida, així que els usuaris no hauran de dedicar el seu temps i els seus recursos en trobar-ne el problema. A canvi, s'ha de mantenir les 24h la màquina virtual on s'executi el software de anàlisi del tràfic així com la web per registrar-se.

Tot i això, es pot implementar la solució en la mateixa màquina que s'està usant actualment per visualitzar el estat de la xarxa i no suposaria un increment molt gran del seu consum.

Finalment, durant el procés de desenvolupament, s'utilitzen els ordinadors personals que ja s'estaven utilitzant anteriorment, pel que no s'estan gastant recursos addicionals i el impacte ambiental.

Un cop acabat el projecte, es pot concloure que la petjada ecològica d'aquest projecte ha estat ínfima ja que com s'ha comentat anteriorment, no s'ha fet ús de cap recurs addicional que no estigués en ús al tractar-se de un desenvolupament software. Durant la vida útil aquesta configuració realitzada es localitzarà en una màquina que ja existeix (la que conté la web que permet mostrar l'estat de la xarxa) i no en augmentarà les despeses en gran mesura.

10.3. Dimensió econòmica

Per determinar si un projecte es viable econòmicament, s'ha de fer un estudi dels costos com s'ha realitzat en l'apartat d'identificació i estimació dels costos. D'aquest estudi es pot veure que la majoria de les despeses ve de recursos humans, que equival al temps que el treballador dedica al projecte. La majoria de les altres despeses es tindrien independentment de la realització del treball, així que el cost del projecte és relativament econòmic.

Al ser un projecte destinat a crear una utilitat per a una comunitat, els beneficiaris en seran els membres. Els permetrà estalviar temps i recursos ja que no hauran de fer l'esforç de detectar si hi ha un error en la xarxa comunitària, sinó que seran notificats directament quan es detecti una fallada que sigui important per a l'usuari.

Un cop acabat el projecte, al haver-se augmentat lleugerament el temps de treball dedicat, el cost també s'ha vist lleugerament augmentat. Tot i això, com s'ha esmentat la majoria dels costos provenen del cost dels recursos humans i l'impacte no és molt gran. El projecte està pensat per a que només calgui la configuració inicial per a funcionar correctament i per tant, no caldria cap manteniment, únicament caldria realitzar modificacions en cas de actualitzar-se la arquitectura o s'hi vulguin introduir millores i un altre cop, només comportaria les despeses del treball humà, i per tant, l'únic cost que tindrà durant la seva vida útil serà el de mantenir el servidor actiu.

10.4. Dimensió social

Pel que fa la dimensió social, tot i no ser una necessitat bàsica, la realització del projecte realment suposarà una millora en la qualitat de vida de tots els usuaris de la xarxa

comunitària de Guifi·Sants, ja que una de les seves preocupacions estarà gestionada automàticament i només s'hauran d'encarregar del problema en el focus d'aquest i en cap cas serà els perjudica, ja que en cas que no estiguin interessats en aquest servei, simplement no en faran ús.

Personalment, la realització del projecte m'ha fet veure que en qualsevol comunitat, indiferentment de lo gran que sigui, sempre es pot col·laborar i intentar prestar ajut o millorar-la, sobretot en una comunitat com Guifi, ja que és una infraestructura creada i mantinguda per els propis usuaris que en fan ús.

11. Conclusions

Per concloure, en aquest projecte s'han aconseguit implementar tots els objectius proposats, que eren:

- Analitzar quines de les dades relacionades amb el tràfic de la xarxa són les més rellevants per a detectar si hi ha alguna fallada en el sistema.
- Dissenyar i implementar un sistema de monitorització i alarma en xarxa, que tingui en compte el estat de la xarxa i detecti les fallades que s'hi produeixin.
- Aconseguir que els usuaris interessats s'hi puguin registrar i sol·licitar les alarmes que els siguin més útils. En cas de que es detecti la fallada que interessa a l'usuari, el sistema de monitorització haurà d'enviar un correu electrònic als usuaris que s'hi hagin subscrit.

I s'ha acabat amb un sistema capaç de gestionar l'estat dels nodes de la xarxa, generant notifikacions cada vegada que no es trobi disponible un node.

Els usuaris tenen una interfície amb la qual es poden registrar i les dades que introdueixen s'introdueixen de forma correcta al sistema, canviant la configuració adequada per tal de que en el mateix moment que envii les dades ja es registri en el sistema i pugui començar a rebre aquestes alarmes.

Per a poder arribar a aquest punt s'han hagut d'usar diverses aplicacions diferents, la majoria de les quals s'ha hagut d'aprendre com funcionen i ha calgut un bon anàlisi per detectar quines funcions calia estudiar amb més deteniment, ja que eren de major rellevància per a la implementació del projecte.

Així doncs, aquest projecte explica com realitzar tota la configuració i com resoldre els problemes que han anat sorgint.

Tot i això hi ha hagut algunes limitacions i el sistema encara té marge de millora. Entre d'altres, es podrien millorar les interfícies, afegir alguna personalització extra de les notifikacions per a cada usuari o agafar les dades dels nodes de forma periòdica en comptes de tenir-ho de forma estàtica.

De cara al futur aquest projecte és molt possible que entri en producció, un cop el director Llorenç Cerdà comprovi la funcionalitat i faci els canvis que consideri, així que es pot dir que s'ha aconseguit elaborar un producte prou bo i que compleix amb les finalitats que s'havien determinat que caldria resoldre.

A nivell personal, és un projecte que m'ha aportat molt, ja que m'ha aportat molts coneixements, no sols tècnics sinó també de metodologia de treball i planificació. Crec que ha estat un projecte que ha permès consolidar molts dels coneixements adquirits durant el transcurs de la carrera i n'estic molt satisfet amb el resultat obtingut.

Referències

- [1] B. R. C.-A. L. M. E. M. R. a. N. L. D. Vega, «A technological overview of the guifi.net community network,» vol. 93, núm. pp. 260 - 278, 2015.
- [2] N. A. a. E. P. L. Cerdà-Alabern, «Experimental Evaluation of Wireless Mesh Networks: A Case Study and Comparison,» núm. NICST'2013, 2013.
- [3] N. A. a. E. P. L. Cerdà-Alabern, «Experimental Evaluation of a Wireless Community,» núm. MSWiM'13, 2013.
- [4] «Guifi.net,» [En línia]. Available: <https://guifi.net/>. [Últim accés: 2 3 2018].
- [5] «Guifi Sants,» [En línia]. Available: <http://sants.guifi.net/>. [Últim accés: 2 3 2018].
- [6] «Google WiFi,» [En línia]. Available: <https://support.google.com/wifi#topic=7216602>. [Últim accés: 6 3 2018].
- [7] «CISCO Meraki,» [En línia]. Available: <http://meraki.cisco.com>. [Últim accés: 6 3 2018].
- [8] «Quick Mesh Project,» [En línia]. Available: <http://qmp.cat..> [Últim accés: 6 3 2018].
- [9] «Nagios,» [En línia]. Available: <https://www.nagios.org/>. [Últim accés: 3 3 2018].
- [10] «qMp Wireless Mesh,» [En línia]. Available: qMp Wireless Mesh. [Últim accés: 1 3 2018].
- [11] «Git,» [En línia]. Available: <https://git-scm.com/>. [Últim accés: 6 3 2018].
- [12] «GitHub,» [En línia]. Available: <https://github.com/>. [Últim accés: 6 3 2018].
- [13] «Google Drive,» [En línia]. Available: <https://drive.google.com>. [Últim accés: 6 3 2018].

Annexos

Annex A javascript per obtenir dades del JSON

```
<!DOCTYPE html>

<html>

<body>


<p>Looping through arrays inside arrays.</p>


<p id="demo"></p>


<script>

var myObj, i, j, x = "";
myObj = codi JSON de la web***

for (i in myObj.nodeList) {
    x += "<h2>" + myObj.nodeList[i].name + "</h2>";
    x += myObj.nodeList[i].data.ipv4[0] + "<br>";

}

document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

***Per a major claredat no s'ha copiat tot el JSON sencer

Annex B codi del cgi

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
# Llibreries
import cgi, os, re, sys, string, time, csv, MySQLdb
# Constants i variables globals
print "Content-type: text/html\n\n"
# Programa principal
form = cgi.FieldStorage()

email = form.getvalue('email')
ip = form.getvalue('ip')
list = []
names = []

db = MySQLdb.connect(host="localhost", # your host
                     user="root",      # username
                     db="prova")       # name of the database

cur = db.cursor()

#miro si ja existeix el usuari a la base de dades
sql = "SELECT * FROM Usuaris WHERE Usuari ="
sql = sql + email + ";"

cur.execute(sql)

#### si existeix, elimino totes les seves referencies

if cur.rowcount != 0:
    sql = "DELETE FROM Usuaris WHERE Usuari ="
    sql = sql + email + ";"
    cur.execute(sql)
    db.commit()

llista_usuaris=[]
cur.execute("SELECT distinct Usuari FROM Usuaris")
items = cur.fetchall()
for row in items:
    if not row[0] in llista_usuaris:
        llista_usuaris.append(row[0])
```

```

#print                                     llista_usuaris
str                                         =                                     ""
for x                                     in llista_usuaris:
    sql = "SELECT IP FROM NodeIP WHERE NomNode IN (SELECT
NomNode FROM Usuaris WHERE Usuari = '" + x
    sql = sql + "');"
# print sql
    str = str + "El usuari " + x + " usa les ips: "
    cur.execute(sql)
    lip = cur.fetchall()
    for y in lip:
        str = str + y[0]
#print str

```

```

list.append(form.getvalue('BCCanBruixa20Rd6'))
names.append('BCCanBruixa20Rd6')
list.append(form.getvalue('BCN-GS-CanBruixa20-NSM5-e48b'))
names.append('BCN-GS-CanBruixa20-NSM5-e48b')
list.append(form.getvalue('BCN-GS-CanBruixa20-RKM5-7bbd'))
names.append('BCN-GS-CanBruixa20-RKM5-7bbd')
list.append(form.getvalue('BCN-GS-JpTarradellas40-6a5f'))
names.append('BCN-GS-JpTarradellas40-6a5f')
list.append(form.getvalue('BCN-GS-Salou2bis-3a56'))
names.append('BCN-GS-Salou2bis-3a56')
list.append(form.getvalue('BCN-NogueraPallaresa15-ac8f'))
names.append('BCN-NogueraPallaresa15-ac8f')
list.append(form.getvalue('BCN-Sagunt66-8971'))
names.append('BCN-Sagunt66-8971')
list.append(form.getvalue('BCNBartomeuPi29-1141'))
names.append('BCNBartomeuPi29-1141')
list.append(form.getvalue('BCNBerlin20-1edf'))
names.append('BCNBerlin20-1edf')
list.append(form.getvalue('BCNCanBatlloBloc11Rd1-1775'))
names.append('BCNCanBatlloBloc11Rd1-1775')
list.append(form.getvalue('BCNEntenca239-0f6b'))
names.append('BCNEntenca239-0f6b')
list.append(form.getvalue('BCNGranVia146Rd1-75a1'))
names.append('BCNGranVia146Rd1-75a1')
list.append(form.getvalue('BCNJardiBotanicSants186-ba35'))
names.append('BCNJardiBotanicSants186-ba35')
list.append(form.getvalue('BCNJocsFlorals148-6a1b'))
names.append('BCNJocsFlorals148-6a1b')
list.append(form.getvalue('BCNMelciorPalau62-6a4c'))
names.append('BCNMelciorPalau62-6a4c')
list.append(form.getvalue('BCNNovell47-39be'))
names.append('BCNNovell47-39be')

```

```

list.append(form.getvalue('BCN0lzinelles59-f0a4'))
names.append('BCN0lzinelles59-f0a4')
list.append(form.getvalue('BCNPuiggari17-b693'))
names.append('BCNPuiggari17-b693')
list.append(form.getvalue('BCNRadas83Rd1-2252'))
names.append('BCNRadas83Rd1-2252')
list.append(form.getvalue('BCNRobrenyo44-4658'))
names.append('BCNRobrenyo44-4658')
list.append(form.getvalue('BCNRobrenyo70-8a35'))
names.append('BCNRobrenyo70-8a35')
list.append(form.getvalue('BCNSagunt62-1667'))
names.append('BCNSagunt62-1667')
list.append(form.getvalue('BCNSanFructuos113-40c3'))
names.append('BCNSanFructuos113-40c3')
list.append(form.getvalue('BCNSantBaltassar25-7fe6'))
names.append('BCNSantBaltassar25-7fe6')
list.append(form.getvalue('BCNSantaCatalina59Rd2-f500'))
names.append('BCNSantaCatalina59Rd2-f500')
list.append(form.getvalue('BCNSantaCatalina59Rd3-805c'))
names.append('BCNSantaCatalina59Rd3-805c')
list.append(form.getvalue('BCNSants186Rd1-993d'))
names.append('BCNSants186Rd1-993d')
list.append(form.getvalue('BCNSants186Rd2-996c'))
names.append('BCNSants186Rd2-996c')
list.append(form.getvalue('BCNSocors2-389c'))
names.append('BCNSocors2-389c')
list.append(form.getvalue('BCNTeatreDelsSentits-4197'))
names.append('BCNTeatreDelsSentits-4197')
list.append(form.getvalue('BCNValladolid22-d906'))
names.append('BCNValladolid22-d906')
list.append(form.getvalue('BCNcrdnyl9Rd1-f705'))
names.append('BCNcrdnyl9Rd1-f705')
list.append(form.getvalue('BCNcrdnyl9Rd2-acae'))
names.append('BCNcrdnyl9Rd2-acae')
list.append(form.getvalue('BCNNevaristoarnus5-8ed7'))
names.append('BCNNevaristoarnus5-8ed7')
list.append(form.getvalue('BCNNevaristoarnus5Rd3-BPi'))
names.append('BCNNevaristoarnus5Rd3-BPi')
list.append(form.getvalue('BCNNevaristoarnus5Rd4-988e'))
names.append('BCNNevaristoarnus5Rd4-988e')
list.append(form.getvalue('BCNNevaristoarnus5Rd9-c239'))
names.append('BCNNevaristoarnus5Rd9-c239')
list.append(form.getvalue('BCNmasnou24Rd1-d594'))
names.append('BCNmasnou24Rd1-d594')
list.append(form.getvalue('BCNmasnou24Rd2-8416'))
names.append('BCNmasnou24Rd2-8416')

```

```

list.append(form.getvalue('BCNpisuerga17-EPR6'))
names.append('BCNpisuerga17-EPR6')
list.append(form.getvalue('BCNpisuerga17-NSM5XW'))
names.append('BCNpisuerga17-NSM5XW')
list.append(form.getvalue('BCNpisuerga17-mANTBox15s'))
names.append('BCNpisuerga17-mANTBox15s')
list.append(form.getvalue('BcnGranVia146-b233'))
names.append('BcnGranVia146-b233')
list.append(form.getvalue('GS-BCNPisuerga20-4b4c'))
names.append('GS-BCNPisuerga20-4b4c')
list.append(form.getvalue('GS-CanVies-b840'))
names.append('GS-CanVies-b840')
list.append(form.getvalue('GS-RamblaBadal140-NS-0fe8'))
names.append('GS-RamblaBadal140-NS-0fe8')
list.append(form.getvalue('GS26gener10-8710'))
names.append('GS26gener10-8710')
list.append(form.getvalue('GSAvMadrid5-nb-9a57'))
names.append('GSAvMadrid5-nb-9a57')
list.append(form.getvalue('GSAvMadrid5-ns-7369'))
names.append('GSAvMadrid5-ns-7369')
list.append(form.getvalue('GSComteGuell152-61ec'))
names.append('GSComteGuell152-61ec')
list.append(form.getvalue('GSConselldeCent12-a1ef'))
names.append('GSConselldeCent12-a1ef')
list.append(form.getvalue('GSConstitucio174-02ed'))
names.append('GSConstitucio174-02ed')
list.append(form.getvalue('GSRajolers10-6e45'))
names.append('GSRajolers10-6e45')
list.append(form.getvalue('GSgV-nb-207c'))
names.append('GSgV-nb-207c')
list.append(form.getvalue('GSgV-rb-fc89'))
names.append('GSgV-rb-fc89')
list.append(form.getvalue('GSgV-rk-0c33'))
names.append('GSgV-rk-0c33')
list.append(form.getvalue('GSgranVia255-db37'))
names.append('GSgranVia255-db37')
list.append(form.getvalue('GSmVictoria-992f'))
names.append('GSmVictoria-992f')
list.append(form.getvalue('GSmVictoria-RK-8da9'))
names.append('GSmVictoria-RK-8da9')
list.append(form.getvalue('HWBellchite-da28'))
names.append('HWBellchite-da28')
list.append(form.getvalue('HWRafaelCampalans29-1edf'))
names.append('HWRafaelCampalans29-1edf')
list.append(form.getvalue('LHAstilla-b1d7'))
names.append('LHAstilla-b1d7')

```

```

list.append(form.getvalue('OpenWrt'))
names.append('OpenWrt')
list.append(form.getvalue('UPC-CN-C6-E104-Alix'))
names.append('UPC-CN-C6-E104-Alix')
list.append(form.getvalue('UPC-CN-C6-E104-Alix-Paul'))
names.append('UPC-CN-C6-E104-Alix-Paul')
list.append(form.getvalue('UPC-CN-C6-E104-NSLM5'))
names.append('UPC-CN-C6-E104-NSLM5')
list.append(form.getvalue('UPC-CN-C6-E104-Portal'))
names.append('UPC-CN-C6-E104-Portal')
list.append(form.getvalue('UPC-CN-C6-E104-Turoffner'))
names.append('UPC-CN-C6-E104-Turoffner')
list.append(form.getvalue('UPC-CN-C6-E206-Alix'))
names.append('UPC-CN-C6-E206-Alix')
list.append(form.getvalue('UPC-CN-C6-E208-Alix-57728'))
names.append('UPC-CN-C6-E208-Alix-57728')
list.append(form.getvalue('UPC-CN-C6-E208-Alix-60415'))
names.append('UPC-CN-C6-E208-Alix-60415')
list.append(form.getvalue('UPC-CN-C6-E208-Alix-92032'))
names.append('UPC-CN-C6-E208-Alix-92032')
list.append(form.getvalue('UPC-CN-C6-Terrat-Alix'))
names.append('UPC-CN-C6-Terrat-Alix')
list.append(form.getvalue('UPC-CN-C6-Terrat-Rocket'))
names.append('UPC-CN-C6-Terrat-Rocket')
list.append(form.getvalue('UPC-CN-D6-105-Rd1'))
names.append('UPC-CN-D6-105-Rd1')
list.append(form.getvalue('UPC-CN-Omega-cf70'))
names.append('UPC-CN-Omega-cf70')
list.append(form.getvalue('UPC-CN-Vertex-Pangea-NS-1193'))
names.append('UPC-CN-Vertex-Pangea-NS-1193')
list.append(form.getvalue('UPC-EETAC-d9f3'))
names.append('UPC-EETAC-d9f3')
list.append(form.getvalue('UPC-ETSEIB-NS-7094'))
names.append('UPC-ETSEIB-NS-7094')

```

```

imprimir          =          "<html><head><title>enviat
correctament</title></head><body><p>          el          email          es:
"+email+"          </p>"

```

```

i                                =                                0
base      =      "INSERT      INTO      Usuaris      VALUES      ('"
mig                                =                                "' , '"
final                                =                                "');"
count                                =                                0
for                                x                                in                                list:
                                if                                not                                (x                                is                                None):

```

```

        count = count + 1
        sql = base + email + mig + names[i] + final
        cur.execute(sql)
        db.commit()
        imprimir = imprimir + "<p> Has seleccionat
el      node      "      +      names[i]+      "</p>"
        i = i+1
if      count      ==      0:
        imprimir = "<p> No has triat cap node, per tant ja
no rebras cap notificacio mes d'aquest sistema"
print      imprimir

res = open("contactes",'r').read()
cur.execute("SELECT distinct Usuari FROM Usuaris")
items = cur.fetchall()
for      row      in      items:
        res = res + " \n define contact{ \n contact_name " +
row[0] + " \n email " + row[0] + " \n
service_notification_options      w,u,c,r,f,s \n
service_notification_commands      notify-service-by-email \n
host_notification_options      d,u,r,f,s \n
host_notification_commands      notify-host-by-email \n}
\n"

cur.execute("SELECT      NomNode      FROM      NodeIP")
items = cur.fetchall()

for      row      in      items:
        sql = "SELECT Usuari FROM Usuaris WHERE NomNode='" +
row[0] + "'"
        cur.execute(sql)
        items2 = cur.fetchall()
        res = res + "define contactgroup { \ncontactgroup_name
" + row[0] + " \nmembers
        for      x      in      items2:
                res = res + x[0] + ", "
                res = res + x[0] + "\n} \n"

open("/usr/local/nagios/etc/objects/contacts.cfg",
'w').write(res)

```

Annex C script per a crear les comandes de Nagios

```

#!/usr/bin/python

import cgi, os, re, sys, string, time, csv

```

```

import math

#with open("nom_lip") as f:
#    lines = f.readlines()
lines = open("nom_lip",'r').read().splitlines()
i = 0

for x in lines:
    if (i % 2 == 0):
        res = "define command { \n command_name " + x +
"_check \n command_line $USER1$/check_ping -H "+ lines[i+1]
+" -w 3000,80% -c 5000,99% -p 5 \n}"
        print res
        i = i + 1

```


Annex D script per crear els serveis de Nagios

```
#!/usr/bin/python

import cgi, os, re, sys, string, time, csv
import math

#with open("nom_lip") as f:
#    lines = f.readlines()
lines = open("nom_lip", 'r').read().splitlines()
i = 0

for x in lines:
    if (i % 2 == 0):
        res = "define service { \nuse local-service
\nhost_name localhost \ncheck_command " + x + "_check
\nservice_description El node " +x+ "no es troba
accessible\ncontact_groups \n}"
        print res
    i = i + 1
```